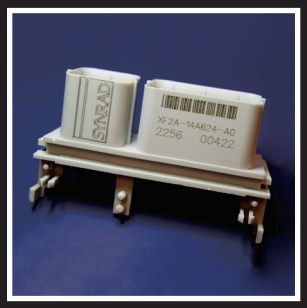
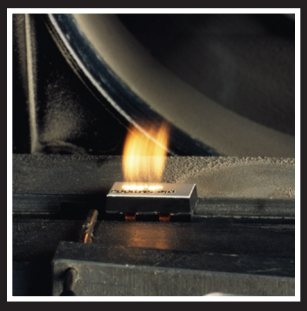
WinMarkpro®

Laser Marking Software





User Guide

SYNRAD®

An Excel Technology Company

4600 Campus Place
Mukilteo, WA 98275
1.800.SYNRAD1
tel 1.425.349.3500
fax 1.425.349.3667
e-mail synrad@synrad.com
www.synrad.com

WinMarkpro®

Laser Marking Software
User Guide

Version 3.0

Released May 2006

Part number 900-17726-02

SYNRAD®

An Excel Technology Company

4600 Campus Place
Mukilteo, WA 98275
1.800.SYNRAD1
tel 1.425.349.3500
fax 1.425.349.3667
e-mail synrad@synrad.com
www.synrad.com

Front Matter

Table of Contents iii

List of Figures xii

List of Tables xvii

Trademark/copyright information xviii

Warranty information xix

Contact information xx

Worldwide headquarters xx

European headquarters xx

Laser Safety

Hazard information LS-1

Terms LS-1

General hazards LS-1

Agency compliance LS-4

Center for Devices and Radiological Health (CDRH) requirements LS-4

Federal Communications Commission (FCC) requirements LS-5

European Union (EU) requirements LS-5

Getting Started section 1

Chapter 1 - Introduction 3

New features 3

System requirements 4

Windows® XP 4

Windows® 2000 4

Windows® NT4 5

Chapter 1 - Introduction (cont.)

Windows® 98 5

Upgrades 5

Version information 5

Version 5.0 (WinXP/2000-PCI compatible) 5

Version 4.0 (WinXP/2000/NT4/98-PCI compatible) 6

Version 3.1 (WinNT4-PCI compatible) 6

Version 2.0 (Win98/95-PCI/ISA/DAC compatible) 6

WinMark Lite 6

Version 1.0 (Win98/95-ISA/DAC compatible) 6

WinMark Pro hardware/software compatibility 7

Mark file compatibility 7

Across operating systems 7

Across WinMark Pro versions 7

Technical support 8

Chapter 2 - Installation 9

Inventory 9

Hardlock installation 10

USB port hardlock 10

Parallel port hardlock 10

WinMark Pro installation 10

WinMark Launcher 11

WinMark Digital Scope 11

Samples folder 11

Fiber Link Controller Card (FLCC) drivers 12

WinMark File Import Filters 12

Chapter 2 - Installation (cont.)

Version 4.0 and later....12
Earlier versions....12
Un-installing WinMark Pro....12

Chapter 3 - Starting WinMark Pro 13

Basic concepts....13
Opening WinMark Pro....13
Changing focusing lens size....14
Setting measurement units....15

Chapter 4 - WinMark Pro Drawing Editor 16

Main Window....16
Toolbar....17
Tool Box....19
Drawing Canvas....21
Slide Bar....21
Object List....21
Property List....22
Help window....22
Status Bar....22

Chapter 5 - Creating a drawing.... 23

Opening a new drawing....23
Adding text....23
Adding bar codes....24
1D bar codes....24
2D codes....24

Chapter 5 - Creating a drawing (cont.)

Adding other objects 24

Rectangles and arcs....24

Spots 24

Importing a graphics file 25

Precision positioning 26

Chapter 6 - File development 27

Off-line mark file development 27

FH Smart mark file development 27

Chapter 7 - WinMark Launcher 28

Introduction....28

Launcher display 29

Progress window....29

Mark Preview window 29

Mark Log window 29

Opening a drawing....30

Marking parts 30

Modifying a drawing 31

Chapter 8 - WinMark Digital Scope 32

WinMark Digital Scope....32

Basic Operation section 35

Chapter 9 - Marking basics 37

Types of marks 38
Text 38
Graphics 41
Bar codes 43
2D codes 44
Marking parameters 44
Velocity 44
Power 44
Resolution 45
Delays 45
Optimization 45
Vector objects 45
Raster objects 46

Chapter 10 - Marking parts 47

The mark session 47
Piece marking 48
Batch marking 49

Chapter 11 - Fast 2D codes 52

Introduction 52
Vector-style 2D codes 53
Spot-style 2D codes 54

Chapter 12 - Automation basics 57

System configuration 57

Additional peripheral cards 57

Additional I/O capability 57

Multiple marking heads 58

Input/output configuration 58

I/O connections 58

Voltage levels 59

Fast Acting Safety Interlock (FASI) feature 59

Developing a basic event sequence 60

Event 0 62

Event 1 63

Event 2 63

Event 3 64

Event 4 64

Automated serialization 65

Serial number setup 65

Marking setup 66

Automation setup 66

Begin automated marking 68

Chapter 13 - ActiveMark automation 70

ActiveMark technology 70

ActiveMark basics 70

Loading the WinMark Pro ActiveMark (ActiveX) control 71

Sample VB and C++ projects 73

Chapter 14 - Custom Test Mark feature 74

Test Mark Control options 75

Test Mark Mode 75

Test Mark Type 75

Creating a custom test mark 76

Setting Test Mark Control options 76

Downloading custom test mark files 77

Custom test mark I/O 77

Chapter 15 - Tracker marking 79

WinMark Pro Tracking tab 79

Determining line speed 84

Definitions 84

Tracker marking criteria 87

Line speed formula 90

Sample calculations 90

Marking text 92

Marking graphics 94

Mark placement 94

Optimization 95

Tracking variables 95

WinMark Pro property variables 95

Summary 95

Commands & Properties section 97

Chapter 16 - WinMark Pro menu commands 99

File 99

Edit 104

Objects 106

View 109

Tools 110

Window 129

Help 130

Chapter 17 - Drawing properties 132

Tracking tab 132

Automation tab 136

Marking tab 139

Format tab 141

All tab 143

Chapter 18 - Object properties 144

Special tab 144

Automation tab 145

Marking tab 152

- Wobble properties 153
- Delay properties 154

Format tab 159

- Array properties 160
- Text format properties 161
- Polyline format properties 164

Chapter 18 - Object properties (cont.)

Arc format properties 165
Bar code format properties....167
2D code format properties 170
Spot format properties 172
Bitmap format properties....173
All tab 177

Chapter 19 - Event Builder commands 178

Event Builder commands....178

Chapter 20 - ActiveMark technology....191

ActiveMark control overview 191
Sample code....192
XY coordinate system 193
ActiveMark control properties....193
ActiveMark control methods 194

Chapter 21 - Troubleshooting....233

Fiber Link Controller Card (FLCC)....233
FH Series marking head....235
Automation....239
Graphics....240
Vector graphics 240
Bitmap graphics 240
ActiveMark....242
Laser Marking FAQ 243
Resources 243

List of Figures

Figure 2-1 WinMark Pro inventory 9

Figure 3-1 Lens Selection dialog box 14
Figure 3-2 Changing "Show Units Type" 15

Figure 4-1 WinMark Pro Main Window 16
Figure 4-2 Toolbar icons 17
Figure 4-3 Tool Box 19
Figure 4-4 Drawing Canvas 21

Figure 5-1 Text Caption Editor dialog box 23
Figure 5-2 Sample imported graphic file 25
Figure 5-3 Object Transformations dialog box 26

Figure 7-1 Launcher window 28

Figure 8-1 Digital Scope display 32

Figure 9-1 Vector versus raster marking 38
Figure 9-2 Font types 39
Figure 9-3 Marking tab object properties 39
Figure 9-4 Family.tif 41
Figure 9-5 Chairbin.dxf 42

Figure 10-1 Mark session flowchart 47
Figure 10-2 Mark button on the toolbar 48
Figure 10-3 Synrad WinMark dialog box 49
Figure 10-4 Setting number of Array Columns 50
Figure 10-5 Setting number of Array Rows 50
Figure 10-6 Mark button on toolbar 51
Figure 10-7 Synrad WinMark dialog box 51

Figure 11-1 Standard raster-filled 2D code 52
Figure 11-2 Vector-style 2D Data Matrix code 53

List of Figures (cont.)

Figure 11-3 Cell size/spot size concept 54
Figure 11-4 Spot-style 2D Data Matrix code 54

Figure 12-1 Start Mark/FASI automation setup 60
Figure 12-2 Sample state diagram 61
Figure 12-3 Setting "Set Digital Before Piece" 62
Figure 12-4 Setting "Wait Digital Before Piece" 63
Figure 12-5 Clearing "Set Digital Before Mark" 63
Figure 12-6 Clearing "Wait Digital After Piece" 64
Figure 12-7 Setting "Set Digital After Mark" 64
Figure 12-8 Serialization setup 66
Figure 12-9 Marking setup 66
Figure 12-10 Automation setup 67
Figure 12-11 Launcher window 68

Figure 13-1 WinMark Pro's ActiveX control in VB Toolbox 71
Figure 13-2 Inserting WinMark Pro's ActiveX control 72
Figure 13-3 Confirm Classes dialog box 73

Figure 14-1 Test Mark Control dialog box 74

Figure 15-1 Tracking tab 79
Figure 15-2 Part sensor output waveform 80
Figure 15-3 Drawing Canvas coordinates relative to Tracker 81
Figure 15-4 Upstream part sensor, rising edge trigger 82
Figure 15-5 Downstream part sensor 82
Figure 15-6 X-Y "start mark" coordinates 83
Figure 15-7 Usable Field Size 85
Figure 15-8 Mark Pitch greater than Usable Field Size 85
Figure 15-9 Tracking Window 86
Figure 15-10 Target Area 87
Figure 15-11 Checking Encoder Resolution 88

List of Figures (cont.)

Figure 15-12 Mark length comparisons 89
Figure 15-13 Maximum marking position in Tracking Window 89
Figure 15-14 Tracker line speed calculation # 1 90
Figure 15-15 Tracker line speed calculation # 2 91
Figure 15-16 Sample text mark 92
Figure 15-17 Beam/mark motion 93
Figure 15-18 Mark orientation 93
Figure 15-19 Configure Marking Order dialog box 94

Figure 16-1 File menu 99
Figure 16-2 Choose Import Type dialog 101
Figure 16-3 Edit menu 104
Figure 16-4 Objects menu 106
Figure 16-5 Align Objects dialog 107
Figure 16-6 Transformations dialog 108
Figure 16-7 View menu 109
Figure 16-8 Tools menu 110
Figure 16-9 Application Settings tab 111
Figure 16-10 Drawing Defaults tab 114
Figure 16-11 Tracker properties on Drawing Defaults tab 115
Figure 16-12 Object Defaults tab 117
Figure 16-13 Stroke / TrueType height comparison 121
Figure 16-14 Startup Options dialog box 125
Figure 16-15 Test Mark Control dialog box 126
Figure 16-16 Window menu 129
Figure 16-17 Help menu 130
Figure 16-18 About Synrad WinMark dialog box 130
Figure 16-19 Marking Head dialog box 130
Figure 16-20 Automation flowchart 131

List of Figures (cont.)

Figure 17-1 Tracking tab drawing properties 132
Figure 17-2 Part sensor output waveform 133
Figure 17-3 Motion Vector coordinates relative to Tracker 133
Figure 17-4 Upstream part sensor, rising edge trigger 134
Figure 17-5 X-Y 'start mark' coordinates 135
Figure 17-6 Automation tab Drawing events 136
Figure 17-7 Marking tab Drawing properties 139
Figure 17-8 Format tab Drawing properties 141
Figure 17-9 All tab Drawing properties 143

Figure 18-1 Special tab object properties 144
Figure 18-2 Automation tab object properties 145
Figure 18-3 Date Format Customization dialog box 147
Figure 18-4 Serial Port Settings dialog box 149
Figure 18-5 Marking tab object properties 152
Figure 18-6 Delay settings 154
Figure 18-7 Pline Start Delay settings 155
Figure 18-8 Pline End Delay settings 156
Figure 18-9 Interseg Delay settings 156
Figure 18-10 Off Vector Delay settings 157
Figure 18-11 Interseg Break Angle settings 158
Figure 18-12 Format tab object properties 159
Figure 18-13 Format tab text properties 161
Figure 18-14 Format tab polyline properties 164
Figure 18-15 Format tab arc properties 165
Figure 18-16 Format tab bar code properties 167
Figure 18-17 Format tab 2D code properties 170
Figure 18-18 Format tab spot properties 172
Figure 18-19 Format tab bitmap properties 173
Figure 18-20 Normal and inverted bitmap images 173

List of Figures (cont.)

Figure 18-21 8-bit black & white image versus 1-bit Error Diffusion reduction.....174
Figure 18-22 8-bit black & white image versus 8-bit Grayscale reduction 175
Figure 18-23 Sample Power/Velocity matrix 176
Figure 18-24 All tab object properties 177

Figure 19-1 Selecting Event Builder commands 178
Figure 19-2 AssignDate command.....179
Figure 19-3 CallDLLFunc command 181
Figure 19-4 MatchDigitalState command 183
Figure 19-5 Event Builder setup for MatchDigitalState demo 185
Figure 19-6 ReadFromSerialPort command 187

Figure 20-1 Sample AutoTestMark form 208

Figure 21-1 Application Settings tab.....234
Figure 21-2 Digital Scope window.....239
Figure 21-3 Bitmap image size/resolution comparison.....241

List of Tables

Table 1-1 Compatibility matrix 7

Table 2-1 Shipping box contents 9

Table 11-1 Lens spot size 55

Table 12-1 Input bit table 58

Table 12-2 Output bit table 59

Table 16-1 Supported file formats 103

Table 17-1 FH Series/Fenix lens specifications 142

Table 17-2 SH/DH lens information 142

Table 18-1 1D bar code formats 168

Trademark/copyright information

SYNRAD and WinMark Pro are registered trademarks of SYNRAD, Inc.

Fenix and ActiveMark are trademarks of SYNRAD, Inc.

All other trademarks or registered trademarks are the property of their respective owners.

© 2006 by SYNRAD, Inc.
All rights reserved.

Warranty information

SYNRAD, Inc. warrants that for a period of ninety (90) days after your receipt of the software, that the disks on which the Software is distributed will be free from defects in material and workmanship under normal operating conditions. SYNRAD will replace defective disks at no charge provided you return the defective disks along with a copy of your receipt to SYNRAD within 90 days of the date you took possession of the software. The warranty is the only express warranty pertaining to this software program and no other representations or claims of any nature shall be binding or obligate SYNRAD, Inc., including implied warranties of merchantability or fitness for a particular period. This limited warranty gives you specific rights. You may have others, which vary from state to state.

SYNRAD, Inc. and SYNRAD Authorized Distributors have the sole authority to make warranty statements regarding SYNRAD products. SYNRAD, Inc. and its Authorized Distributors neither assumes nor authorizes any representative or other person to assume for us any other warranties in connection with the sale, service, or shipment of our products. SYNRAD, Inc. reserves the right to make changes and improvements in the design of our products at any time without incurring any obligation to make equivalent changes in products previously manufactured or shipped. Buyer agrees to hold SYNRAD harmless from any and all damages, costs, and expenses relating to any claim arising from the design, manufacture, or use of the product, or arising from a claim that such product furnished Buyer by SYNRAD, or the use thereof, infringes upon any Patent, foreign or domestic.

Single user, one computer license

SYNRAD, Inc. grants you a non-exclusive "single-user" license to use our computer application (the "Software") and the related documentation according to the terms set forth in this agreement. SYNRAD, Inc. reserves all rights, including ownership of all copies of this software not expressly granted to you.

You may:

a. Use the Software on only one computer at time.

b. Have the Software installed on only one computer at a time (this means you need to purchase two copies if you wish to run it at home and at work.

c. Make one backup copy of the Software, which is automatically subject to this agreement, in support of your use of this Software.

d. Transfer the Software, including all copies and Documentation, to another party if the other party agrees to accept the terms and conditions of this agreement. Your transfer of the Software terminates your license from SYNRAD , Inc.

You may not:

a. Distribute the Software, Documentation, Photographs, or copies or portions thereof in any manner or form or use.

b. Copy or transfer the Software. Documentation, or Photographs, in whole or in part, except as provided above.

c. Duplicate or electronically distribute the Software. Documentation, or Photographs, except as provided above.

d. Alter, merge, modify, adapt, decompile, or reverse engineer the Software.

e. Remove, obscure, or alter any SYNRAD, Inc. patent, copyright, or other proprietary rights notice.

Contact information

Worldwide headquarters

SYNRAD's worldwide headquarters are located north of Seattle in Mukilteo, Washington, U.S.A. Our mailing address is:

SYNRAD, Inc.
4600 Campus Place
Mukilteo, WA 98275 U.S.A.

Phone us at:	1.800.SYNRAD1 (1.800.796.7231)
Outside the U.S.:	+1.425.349.3500
Fax:	+1.425.349.3667
Email:	synrad@synrad.com

Sales and Applications

SYNRAD's Regional Sales Managers work with customers to identify and develop the best CO_2 laser solution for a given application. Because they are familiar with you and your laser application, use them as a first point of contact when questions arise. Regional Sales Managers also serve as the liaison between you and our Applications Lab in processing material samples per your specifications.

Customer Service

For assistance with order/delivery status, service status, or to obtain a Return Authorization (RA) number, contact SYNRAD and ask to speak to a Customer Service representative.

Technical Support

SYNRAD's Regional Sales Managers are able to answer many technical questions regarding the installation, use, troubleshooting, and maintenance of our products. In some cases, they may transfer your call to a Laser, Marking Head, or Software Support Specialist. You may also email questions to the Technical Support Group by sending your message to support@synrad.com or to support@winmark.com.

Reference materials

Your Regional Sales Manager can provide reference materials including Outline & Mounting drawings, Operator's Manuals, Technical Bulletins, and Application Newsletters. Most of these materials are also available directly from SYNRAD's web site at http://www.synrad.com.

European headquarters

SYNRAD's European subsidiary, Excel Technology Europe GmbH, covers Austria, Germany, and Italy. Contact Excel Technology at:

Excel Technology Europe GmbH
Münchner Str. 2a
D-82152 Planegg
Germany

Phone:	+49 89 891 462-0
Fax:	+49 89 891 462-69

Hazard information

Hazard information includes terms, symbols, and instructions used in this manual or on the equipment to alert both operating and service personnel to the recommended precautions in the care, use, and handling of Class IV laser equipment.

Terms

Certain terms are used throughout this manual or on the equipment labels. Please familiarize yourself with their definitions and significance.

Danger: Imminent hazards which, if not avoided, will result in death or serious injury.

Warning: Potential hazards which, if not avoided, could result in death or serious injury.

Caution: Potential hazards or unsafe practices which, if not avoided, may result in minor or moderate injury.

Caution: Potential hazards or unsafe practices which, if not avoided, may result in product damage.

Note: Points of particular interest for more efficient or convenient equipment operation; additional information or explanation concerning the subject under discussion.

General hazards

Following are descriptions of general hazards and unsafe practices that could result in death, severe injury, or product damage. Specific warnings and cautions not appearing in this section are found throughout the manual.

Danger
serious personal injury

SYNRAD CO_2 laser products emit ***invisible*** infrared laser radiation in the 10.6 µm CO_2 wavelength band.

Do not allow laser radiation to enter the eye by viewing direct or reflected laser energy. CO_2 laser radiation can be reflected from metallic objects even though the surface is darkened. Direct or diffuse laser radiation can inflict severe corneal injuries leading to permanent eye damage or blindness. All personnel must wear eye protection suitable for 10.6 µm CO_2 radiation when in the same area as an exposed laser beam. Eyewear protects against scattered energy but is not intended to protect against direct viewing of the beam—never look directly into the laser output aperture or view scattered laser reflections from metallic surfaces.

Enclose the beam path whenever possible. Exposure to direct or diffuse CO_2 laser radiation can seriously burn human or animal tissue, which may cause permanent damage.

Danger
serious personal injury

SYNRAD CO_2 laser products are not intended for use in explosive, or potentially explosive, atmospheres.

Warning
serious personal injury

U.S. customers should refer to and follow the laser safety precautions described in the American National Standards Institute (ANSI) Z136.1-2000 document, *Safe Use of Lasers*. Procedures listed in this Standard include the appointment of a Laser Safety Officer (LSO), operation of the product in an area of limited access by trained personnel, servicing of equipment only by trained and authorized personnel, and posting of signs warning of the potential hazards.

European customers should appoint a Laser Safety Officer (LSO) who should refer to and follow the laser safety precautions described in EN 60825-1, 2001 – *Safety of Laser Products*.

Warning
serious personal injury

Materials processing with a laser can generate air contaminants such as vapors, fumes, and/or particles that may be noxious, toxic, or even fatal. Material Safety Data Sheets (MSDS) for materials being processed should be thoroughly evaluated and the adequacy of provisions for fume extraction, filtering, and venting should be carefully considered. Review the following references for further information on exposure criteria:

ANSI Z136.1-2000, *Safe Use of Lasers*, section 7.3.

U.S. Government's *Code of Federal Regulations*: 29 CFR 1910, Subpart Z.

Threshold Limit Values (TLV's) published by the American Conference of Governmental Industrial Hygienists (ACGIH).

It may be necessary to consult with local governmental agencies regarding restrictions on the venting of processing vapors.

Warning
serious personal injury

The use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

SYNRAD CO_2 laser products should be installed and operated in manufacturing or laboratory facilities by trained personnel only. Due to the considerable risks and hazards associated with the installation and operational use of any equipment incorporating a laser, the operator must follow product warning labels and instructions to the user regarding laser safety.

Exercise safe operating practices per ANSI Z136.1-2000 at all times when actively lasing.

To prevent exposure to direct or scattered laser radiation, follow all safety precautions specified throughout this manual as well as precautions given in laser and marking head Operator's Manuals.

Do not place your body or any combustible object in the path of the laser beam.

Always wear safety glasses or protective goggles with side shields to reduce the risk of damage to the eyes when operating the laser.

A CO_2 laser is an intense heat source and will ignite most materials under proper conditions. Never operate the laser in the presence of flammable or explosive materials, gases, liquids, or vapors.

The use of controls or adjustments or performance of procedures other than those specified herein may result in exposure to hazardous ***invisible*** laser radiation, damage to, or malfunction of the laser. Severe burns will result from exposure to the laser beam.

Safe operation of the laser requires the use of an external beam block to safely block the beam from traveling beyond the desired work area. Use a water-cooled beam dump or power meter, or similar non-scattering, noncombustible material as the beam block. Never use organic material or metals as the beam blocker; organic materials, in general, are apt to combust or melt and metals act as specular reflectors which may create a serious hazard outside the immediate work area.

Other hazards

The following hazards would be typical for SYNRAD CO_2 laser products when incorporated for intended use: (A) risk of injury when lifting or moving the unit; (B) risk of exposure to hazardous laser energy through unauthorized removal of access panels, doors, or protective barriers; (C) risk of exposure to hazardous laser energy and injury due to failure of personnel to use proper eye protection and/or failure to adhere to applicable laser safety procedures; (D) risk of exposure to hazardous or lethal voltages through unauthorized removal of covers, doors, or access panels; (E) generation of hazardous air contaminants that may be noxious, toxic, or even fatal.

Additional laser safety information

The SYNRAD web site (http://www.synrad.com/LaserFacts/lasersafety.html) contains an online laser safety handbook that provides information on (1) Laser Safety Standards for OEM's/System Integrators including product classification, product housing, product features, and other CDRH requirements; (2) Laser Safety Standards for End Users including Class I installations, Class IV installations, laser hazards, ANSI Standard, U.S. State requirements, and OSHA; (3) References and Sources including CDRH, ANSI/OSHA; and (4) Assistance with Requirements.

In addition, the Occupational Safety and Health Administration (OSHA) provides an online Technical Manual (located at http://www.osha-slc.gov/dts/osta/otm/otm_iii/otm_iii_6.html). Section III, Chapter 6 and Appendix III are good resources for laser safety information. Another excellent laser safety resource is the Laser Institute of America (LIA). Their comprehensive web site is located at www.laserinstitute.org.

Agency compliance

The *Agency compliance* section includes subsections:

- Center for Devices and Radiological Health (CDRH) requirements
- Federal Communications Commission (FCC) requirements
- European Union (EU) requirements

SYNRAD lasers are designed, tested, and certified to comply with certain United States (U.S.) and European Union (EU) regulations. These regulations impose product performance requirements related to electromagnetic compatibility (EMC) and product safety characteristics for industrial, scientific, and medical (ISM) equipment. The specific provisions to which systems containing SYNRAD lasers must comply are identified and described in the following paragraphs. Note that compliance to CDRH, FCC, and EU requirements depends in part on the laser version selected—Keyswitch or OEM.

In the U.S., laser safety requirements are governed by the Center for Devices and Radiological Health (CDRH) under the auspices of the U.S. Food and Drug Administration (FDA) while radiated emission standards fall under the jurisdiction of the U.S. Federal Communications Commission (FCC). Outside the U.S., laser safety and emissions are governed by European Union (EU) Directives and Standards.

In the matter of CE-compliant laser products, SYNRAD, Inc. assumes no responsibility for the compliance of the system into which the product is integrated, other than to supply and/or recommend laser components that are CE marked for compliance with applicable European Union Directives.

Because OEM laser products are intended for incorporation as components in a laser processing system, they do not meet all of the Standards for complete laser processing systems as specified by 21 CFR, Part 1040 or EN 60825-1. SYNRAD, Inc. assumes no responsibility for the compliance of the system into which OEM laser products are integrated.

Center for Devices and Radiological Health (CDRH) requirements

Keyswitch models

SYNRAD Keyswitch model lasers comply with requirements for Class IV laser products imposed by the Radiation Control for Health and Safety Act of 1968. Under this Act, the U.S. Food and Drug Administration (FDA) issued a performance standard in the *Code of Federal Regulations* (CFR) for laser products. This performance standard, (21 CFR, Subchapter J, Part 1040.10) was developed to protect public health and safety by imposing requirements upon manufacturers of laser products to provide an indication of the presence of laser radiation, to provide the user with certain means to control radiation, and to assure that all personnel are adequately warned of potential hazards through the use of product labels and instructions.

Product features incorporated into the design of SYNRAD Keyswitch lasers to comply with CDRH requirements are integrated as panel controls or indicators, internal circuit elements, or input/output signal interfaces. Specifically, these features include a keyswitch (Keyswitch versions), lase and laser ready indicators, remote interlock for power on/off, a laser aperture shutter switch, and a five-second delay between power on and lasing. Incorporation of certain features is dependent on the laser version (Keyswitch or OEM).

OEM models

SYNRAD OEM lasers are OEM products intended for incorporation as components in laser processing systems. As supplied by SYNRAD, these lasers do not meet the requirements of 21 CFR, Subchapter J without additional safeguards. In the U.S., the Buyer of these OEM laser components is solely responsible for the assurance that the laser processing system sold to an end user complies with all laser safety requirements before the actual sale of the system. Under CDRH regulations, the Buyer must submit a report to the CDRH prior to shipping the system. In jurisdictions outside the U.S., it is the sole responsibility of the Buyer of these OEM components to ensure that they meet all applicable local laser safety requirements. In cases where the Buyer is also the end-user of the OEM laser product, the Buyer/end-user must integrate the laser so that it complies with all applicable laser safety standards as set forth above.

Federal Communications Commission (FCC) requirements

The United States Communication Act of 1934 vested the Federal Communications Commission (FCC) with the authority to regulate equipment that emits electromagnetic radiation in the radio frequency spectrum. The purpose of the Communication Act was to prevent harmful electromagnetic interference (EMI) from affecting authorized radio communication services. The FCC regulations that govern industrial, scientific, and medical (ISM) equipment are fully described in 47 CFR, Part 18, Subpart C.

FCC caution to the user

The Federal Communications Commission warns the user that changes or modifications of the unit not expressly approved by the party responsible for compliance could void the user's authority to operate the equipment.

European Union (EU) requirements

Laser safety standards

Under the Low Voltage Directive, 73/23/EEC, the European Norm (EN) document EN 60825-1 was developed to protect persons from laser radiation by imposing requirements upon manufacturers of laser products to provide an indication of laser radiation; to classify laser products according to the degree of hazard; to require both user and manufacturer to establish procedures so that proper precautions are adopted; to ensure adequate warning of the hazards associated with accessible radiation through signs, labels, and instructions; to improve control of laser radiation through protective features; and to provide safe usage of laser products by specifying user control measures.

Keyswitch models

SYNRAD Keyswitch models are designed to comply with the requirements imposed by EN 60825-1 for Class IV laser products.

OEM models

SYNRAD OEM lasers are OEM products intended for incorporation as components in laser processing systems. As supplied by SYNRAD, these lasers do not meet the requirements of EN 60825-1 without additional safeguards. European Union Directives state that "OEM laser products which are sold to other manufacturers for use as components of any system for subsequent sale are not subject

to this Standard, since the final product will itself be subject to the Standard." This means that Buyers of OEM laser components are solely responsible for the assurance that the laser processing system sold to an end-user complies with all laser safety requirements before the actual sale of the system. Note that when an OEM laser component is incorporated into another device or system, the entire machinery installation may be required to conform to EN 60204/292, *Safety of Machinery*; the Machinery Directive EN 89/392/EEC; and/or any other applicable Standards. In cases where the Buyer is also the end-user of the OEM laser product, the Buyer/end-user must integrate the laser so that it complies with all applicable laser safety standards as set forth above.

Electromagnetic interference standards

The European Union's Electromagnetic Compatibility (EMC) Directive, 89/336/EEC, is the sole Directive developed to address electromagnetic interference (EMI) issues in electronic equipment. In particular, the Directive calls out European Norm (EN) documents that define the emission and immunity standards for specific product categories. For SYNRAD lasers, EN 55011 and CISPR:11 define radiated and conducted RF emission limits while the generic Standards EN 61326 and EN 50082-1 define immunity requirements published by the International Electromechanical Commission (IEC).

When integrating SYNRAD's OEM lasers, the Buyer and/or integrator of the end system is responsible for meeting all applicable Standards to obtain the CE mark.

Getting Started

Use information in this section to install WinMark Pro software and to create a sample drawing for laser marking.

This section contains the following chapters:

- Chapter 1 – Introduction
  describes new features, computer system requirements, software upgrades, version information, mark file compatibility, and technical support availability.
- Chapter 2 – Installation
  explains how to install the Hardlock and WinMark Pro laser marking software.
- Chapter 3 – Starting WinMark Pro
  explains basic concepts and initial setup of operating parameters.
- Chapter 4 – WinMark Pro Drawing Editor
  describes WinMark Pro's drawing environment.
- Chapter 5 – Creating a drawing
  steps through the creation of a sample drawing.
- Chapter 6 – File development
  explains offline and FH Smart file development.
- Chapter 7 – WinMark Launcher
  describes opening and marking files using WinMark Launcher.
- Chapter 8 – WinMark Digital Scope
  describes using Digital Scope to test and troubleshoot laser control and I/O components.

1 Introduction

The *Introduction* chapter includes subsections:

- New features
- System requirements
- Upgrades
- Version information
- WinMark Pro hardware/software compatibility
- Mark file compatibility
- Technical support

Welcome to SYNRAD's WinMark Pro® Laser Marking Software—software that gives you the power to create mark files containing a variety of objects including text, bar code, and even imported professional graphics for laser marking your products. In addition to creating virtually any image, WinMark Pro is capable of controlling every aspect of the laser marking process including incrementing serial numbers, changing laser parameters, and interfacing with automated parts handling equipment.

New features

WinMark Pro software supports SYNRAD's FH Series Index, Tracker, and Smart marking heads; Fenix™ Laser Markers; and is fully compatible with previous DH Series marking heads. New software features found in version 4.0 and later include:

- Vector-style 2D codes
- Multiple language support
- Extended character sets for stroke fonts
- Improved bitmap import functionality
- "Explode" and edit imported vector objects
- Eight-bit (256 level) Grayscale marking
- On-line link to our *Laser Marking FAQ*
- v5.0 introduces SCANLAB marking head support and direct download to FH Smart

Please review the *Release Notes* included with your copy of WinMark Pro. The *Release Notes* describe new features added since the release of this manual.

Check the WinMark Pro web site frequently for the very latest program updates and releases.

System requirements

Note: WinMark Pro versions 4.0 and later requires a PCI-bus Fiber Link Controller Card (FLCC) in order to drive FH Series Index and Tracker marking heads and Fenix Laser Markers. No FLCC is required to communicate with FH Smart heads. A SCANLAB RTC® PC interface board (PCI-bus) is required when using WinMark Pro v5.0 and later to drive SCANLAB marking heads.

In order to install and run WinMark Pro software, your Microsoft® Windows® computer system must meet the following minimum requirements:

Note: WinMark Pro version 4.0 or later software is NOT compatible with Windows® 95 and is NOT compatible with computers using an ISA-bus FLCC.

Windows® XP

- Pentium-compatible CPU, 233 MHz or faster (300 MHz recommended)
- 64-MB RAM (128 MB recommended)
- 20-MB available hard disk space
- One CD-ROM drive
- Super VGA or higher-resolution display (16-bit or 24-bit SVGA color monitor recommended)
- One open PCI-bus slot

Windows® 2000

- Pentium-compatible CPU, 133 MHz or faster
- 64-MB RAM
- 20-MB available hard drive space
- One CD-ROM drive
- VGA or higher-resolution display (16-bit or 24-bit SVGA color monitor recommended)
- One open PCI-bus slot

Windows® NT4

- Pentium-compatible CPU, 90 MHz or faster
- 16-MB RAM (32 MB recommended)
- 20-MB available hard drive space
- One CD-ROM drive
- VGA or higher-resolution display (16-bit or 24-bit SVGA color monitor recommended)
- One open PCI-bus slot

Windows® 98

- 486DX, 66 MHz or faster (Pentium 90 MHz or faster recommended)
- 16-MB RAM (24 MB recommended)
- 20-MB available hard drive space
- One CD-ROM drive
- VGA or higher-resolution display adaptor (16-bit or 24-bit SVGA color monitor recommended)
- One open PCI-bus slot

Upgrades

Unlike many other software packages that cost hundreds of dollars to upgrade, WinMark Pro upgrades are made available free of charge on the WinMark Pro web site. Users should periodically check our web site at: http://www.winmark.com to download the latest released version of WinMark Pro laser marking software.

Version information

WinMark Pro version numbers (1.2.3.××××) and build numbers (×.×.×.1234) are used to track the introduction of new features. A detailed *Version History* is available from the WinMark Pro web site.

Version 5.0

(WinXP/2000-PCI compatible)

Version 5.0 allows WinMark Pro (on Windows XP, 2000, NT4, and 98 operating systems) to operate SYNRAD FH Series marking heads using PCI-bus Fiber Link Controller Cards and adds support for SCANLAB marking heads using a SCANLAB PCI-bus RTC PC interface board. Version 5.0 also provides direct serial communication to FH Series Smart marking heads.

Version 4.0

(WinXP/2000/NT4/98-PCI compatible)

Version 4.0, beginning with the release of 4.0.0.3773, extends WinMark Pro functionality to Windows XP, 2000, NT4, and 98 operating systems using PCI-bus Fiber Link Controller Cards. Windows 95 operating systems and ISA-bus FLCCs are not supported by version 4.0.

Version 3.1

(WinNT4-PCI compatible)

Version 3.1.0.3767 (as well as build 2.1.0.3509NT) allow WinMark Pro to operate on Windows NT4. These builds are compatible only with PCI-bus FLCCs; ISA-bus Fiber Link Cards are not supported.

Version 2.0

(Win98/95-PCI/ISA/DAC compatible)

WinMark Pro versions 2.0.0.3087 through 2.1.0.3767 add support for FH Series Tracker marking heads on Windows 98/95 operating systems. Beginning with build 3468 (v2.1.0.3468), WinMark Pro supports PCI-bus FLCCs.

WinMark Lite

WinMark Lite was packaged with early Fenix Laser Markers, providing an inexpensive method for developing and marking simple files where automation features such as I/O capability and serialization were not required. Version 2.1.0.3767 is the last software build to support WinMark Lite.

Version 1.0

(Win98/95-ISA/DAC compatible)

WinMark Pro versions 1.0.0.1150 through 1.0.3.2055 support both DH and SH Series marking heads running on Windows 98 and 95 operating systems. These versions of WinMark Pro drive the marking head using an ISA-bus Fiber Link Controller Card (DH Series) or a DDA06 Digital-to-Analog card (SH Series). WinMark Pro cannot drive SH Series marking heads using the DADIO DAC card.

Version 1.0.5.2625 through 1.0.6.2856 add support for Fenix Laser Markers and FH Series Index marking heads on Windows 95/98 operating systems using an ISA-bus Fiber Link Controller Card (FLCC).

WinMark Pro hardware/software compatibility

The compatibility matrix shown in Table 1-1 below illustrates the compatibility of WinMark Pro with various computer operating systems and Fiber Link Controller Card types.

Table 1-1 Compatibility matrix

	WinMark Pro (< v2.0.0.3155)	WinMark Pro (v2.1.0.xxxx)	WinMark Pro NT (v3.0.0.xxxx)	WinMark Pro (4.x.x.xxxx)	WinMark Pro (v5.0.x.xxxx)
Platform					
Windows XP	No	No	No	Yes	Yes
Windows 2000	No	No	No	Yes	Yes
Windows NT4	No	No	Yes	Yes	Yes*
Windows 98	Yes	Yes	No	Yes*	Yes*
Windows 95	Yes	Yes	No	No	No
FLCC Type					
PCI-bus	No	Yes	Yes	Yes	Yes
RTC (ScanLab)	No	No	No	No	Yes
ISA-bus	Yes	Yes	No	No	No
DAC	Yes	Yes	No	No	No

* not recommended for optimum performance

Mark file compatibility

Across operating systems

WinMark Pro .mkh mark files are compatible across all supported Windows operating systems.

Across WinMark Pro versions

All WinMark Pro .mkh mark files are forward compatible—files created with older versions will run in newer versions of software. Mark files are not backwards compatible—files created in newer versions may not function properly when run with older versions of WinMark. The key is the build number. Mark files created with software build numbers less than or equal to the build number of the current marking software will always open and mark.

When using WinMark Pro to open mark files created with earlier software versions (prior to build 1.0.6.2856), Pline Start, Pline End, Interseg, and Off Vector delays may need to be modified due to the development of an improved delay algorithm in build 2856.

Technical support

For assistance with many common software and hardware issues, check our *Laser Marking FAQ* on the WinMark Pro web site at http://www.winmark.com or on the WinMark Pro CD.

For in-depth software or marking head support, contact the Technical Support Group via email at support@winmark.com or support@synrad.com.

By phone, contact us at 1.800.SYNRAD1 (or at 1.425.349.3500) between the hours of 8:30 A.M. to 5:00 P.M. Pacific Standard Time.

2 Installation

The *Installation* chapter includes subsections:

- Inventory
- Hardlock installation
- WinMark Pro installation
- WinMark File Import Filters
- Un-installing WinMark Pro

Inventory

The WinMark Pro software package includes the following items:

Table 2-1 Shipping box contents

Shipping Box Contents	Qty
WinMark Pro Software CD	1
Hardlock (USB or parallel port)	1
WinMark Pro User Guide CD	1

Figure 2-1 WinMark Pro inventory

Hardlock installation

A *Hardlock* is installed on the marking computer to prevent unauthorized use of WinMark Pro software. If the *Hardlock* is not installed, WinMark Pro is fully functional except that marking output is disabled. In this "demo" or off-line mode, WinMark Pro is still able to develop, modify, and save new or existing mark files.

Note: FH Smart marking heads do not require a *Hardlock* in order to operate. Files are created and saved in WinMark's "demo" mode and then downloaded to FH Smart via SmartTerm or WinMark v5.0.

USB port hardlock

To install the USB port *Hardlock*, perform the following steps:

1. Locate the *Hardlock* included with the materials shipped with your WinMark Pro purchase.
2. Connect the *Hardlock* to your computer's USB port.

Parallel port hardlock

To install the parallel port *Hardlock*, perform the following steps:

1. Locate the *Hardlock* included with the materials shipped with your WinMark Pro purchase.
2. Disconnect the printer cable (if installed) from the back of your computer.
3. Connect the *Hardlock* to the computer's parallel port.
4. Connect the printer cable (if installed) to the *Hardlock*.

WinMark Pro installation

To install WinMark Pro laser marking software on your computer, perform the following steps:

Important Note: You should install WinMark Pro before installing the Fiber Link Controller Card.

Note: Before upgrading to a newer version of WinMark Pro, backup your existing .mkh mark files and then uninstall any previous versions of WinMark Pro.

1. Locate the *WinMark Pro CD* included with the materials shipped with your WinMark Pro purchase.
2. Exit all programs.
3. Insert the *WinMark Pro CD* into the CD-ROM drive.

   If the CD-ROM Autostart feature is enabled on your computer, then your Internet browser will launch and open the "Welcome to SYNRAD on CD!" page; otherwise, use Windows Explorer to navigate through the CD-ROM and open *TitlePage.htm*.

4. Click the *WinMark Pro software* link.

5 On the *WinMark Pro Laser Marking Software* page, find the version you wish to install and then click the *ReadMeFirst* link to review important installation notes.

Note: On Windows XP, 2000, and NT4 operating systems, you must log on as the Administrator in order to install WinMark Pro and the associated FLCC device drivers. Because of this behavior, WinMark Pro's registry access permissions are initially set to Administrator by the WinMark installer when it sets the startup language. The installer does not support different access levels for registry keys so these must be changed by the program. It is therefore necessary to run WinMark Pro once as an Administrator before other, more restricted, users can run the program.

6 Click the *Install WinMark Pro* link to install the appropriate version for your computer's operating system.

7 Follow the on-screen instructions.

Important Note: During installation, WinMark Pro (version 4.0 and later) presents an option to preserve your existing WinMark settings in the computer's *System Registry*. Choose *Yes* to preserve current WinMark Pro registry settings including custom date code formats, object defaults, shift codes, auto startup info, and saved serial numbers. Choose *No* to erase all previous WinMark registry entries and perform a clean installation.

Before installing earlier versions of WinMark, check our *Laser Marking FAQ* on the WinMark Pro web site (http://www.winmark.com) or the *WinMark Pro CD* for details about saving and restoring registry information.

WinMark Launcher

As part of the software installation, WinMark Launcher (*Launcher.exe*) is placed in the *Program Files\WinMark* folder and a shortcut is placed on the desktop. Launcher allows operators to load existing .mkh files for marking without opening WinMark's Drawing Editor. Using WinMark Pro's Startup Options menu, Launcher can be configured to load a specific file and to start marking automatically when the computer is powered up.

WinMark Digital Scope

Digital Scope (*DigScope.exe*) is a stand-alone program included in the WinMark folder. Use Digital Scope to test individual inputs and outputs, turn the laser beam on or off, or adjust the laser's output power level. This "manual" control is especially useful when testing or troubleshooting I/O connections to the marking head or Fiber Link Controller Card.

Samples folder

A *Samples* folder is also placed in the WinMark folder during installation. This folder contains a sample communications DLL, Microsoft® Visual Basic® and Microsoft® Visual C++™ sample code (demonstrating the use of WinMark's ActiveMark™ control methods), as well as several bitmap and vector graphics files that you can import.

Fiber Link Controller Card (FLCC) drivers

PCI-bus FLCC device drivers are automatically installed by WinMark Pro versions 3.1 and later. Restarting the computer after installation of the Fiber Link Card and software causes the Windows Device Manager (or NT Service) to detect new hardware (the PCI-bus FLCC) and load the correct device driver files. NT4 operating systems must be rebooted again to start the new NT Service.

Customers using WinMark Pro versions 2.1.0.3468 through 2.1.0.3767 must locate the PCI Device Driver files (*Synmarkh.inf* and *Synmarkh.vxd*) on the *WinMark Pro CD*, on Disk 2 of 2 of older two disk installation sets, or copy them from the WinMark Pro web site. After installing the PCI card and rebooting, the Windows *New Hardware* wizard guides you through the driver installation process.

Note: Refer to the *FH Series Marking Head Operator's Manual* or the *Fenix Laser Marker Operator's Manual* for detailed instructions on installing SYNRAD's Fiber Link Controller Card.

Note: ISA-bus FLCC and DAC cards do NOT require drivers, they are direct address devices.

WinMark File Import Filters

Version 4.0 and later

WinMark Pro's File Import Filters, automatically load during installation of WinMark Pro versions 4.0.0.4018 and above. These filters allow you to import a wide variety of graphic file formats. See Chapter 16, *WinMark Pro menu commands*, in the "Commands & Properties" section for a complete list of supported import file types.

Earlier versions

When using earlier versions of WinMark Pro, remember to install File Import Filters from the *WinMark Pro CD* or download and install them from the WinMark Pro web site before importing graphics. To install WinMark's File Import Filters, perform the following steps:

1. Locate the File Import Filter executable file on the *WinMark Pro CD* (on the *WinMark Pro Laser Marking Software* page) or on the WinMark Pro web site.
2. Click the *Install File Import Filters* link to install the graphics import filters.
3. Follow the on-screen instructions.

Un-installing WinMark Pro

If you decide to remove WinMark Pro software from your computer, perform the following steps:

1. Click the *Start* button on the taskbar.
2. Select *Programs* and then locate the *Synrad WinMark* folder.
3. Click the *Uninstall Synrad WinMark* icon.
4. Follow the on-screen instructions.

3 Starting WinMark Pro

The *Starting WinMark Pro* chapter includes subsections:

- Basic concepts
- Opening WinMark Pro
- Changing focusing lens size
- Setting measurements units

Basic concepts

A mark file is created by placing text, barcodes, polyline objects, or imported graphics into a drawing. Each individual object, including the *Drawing Canvas* itself, has its own unique set of properties whose default settings can be modified by the user. Each class of objects has specific sets of properties where each property defines some characteristic of that object; for example, all text objects have a Text Height property. In addition, all objects share a set of general properties that define their position and how the laser transfers that object's image to the workpiece. This ability to precisely control marking output by customizing drawing and object properties is one of the strengths of WinMark Pro laser marking software.

In automated marking systems, ultimate control is provided by WinMark Pro's ActiveMark™ technology. Adding WinMark Pro's ActiveMark (ActiveX®) controls to your Microsoft Visual Basic or MS Visual C++ program offers virtually unlimited control of the marking process. See Chapter 13, *ActiveMark automation*, in the "Basic Operation" section for an introduction to ActiveMark technology.

Opening WinMark Pro

Open WinMark Pro by double-clicking the *Shortcut to WinMark* icon ( ) on your desktop or perform the following steps:

1. Ensure that other WinMark programs including WinMark Launcher and Digital Scope are closed.
2. Click *Start* on the taskbar.
3. Select *Programs* and then locate the *Synrad WinMark* folder.
4. Click the *Synrad WinMark* icon.

Note: If a user password has been set, then the *Enter Password* dialog box opens. Type the correct case-sensitive password and click *OK*.

5. The first time WinMark Pro opens after a new installation, the *Lens Selection* dialog box appears and prompts you to enter the size of the currently installed focusing lens. The lens size entered here determines overall dimensions of the *Drawing Canvas*.

   Double-click the desired lens or make a lens selection and then click *OK*.

Changing focusing lens size

If you change the focusing lens on your marking head to a lens having a different field size, then you will need to enter the new lens size so that WinMark Pro properly scales the drawing when marking.

To change lens settings, perform the following steps:

1 From the *Tools* menu, select General Settings….

2 In the *Applications Settings* dialog box, click the *Application Settings* tab.

3 Click Lens, and then click the ellipsis (…).

4 In the *Lens Selection* dialog box (Figure 3-1), select the lens matching the one currently installed on your marking head.

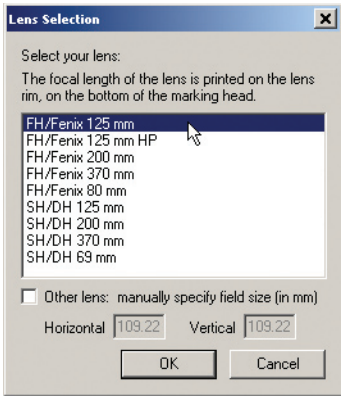


Figure 3-1 Lens Selection dialog box

Note: When using an FH Series marking head or Fenix Laser Marker, select a lens from the FH/Fenix lens group only.

5 Open a new drawing to display a *Drawing Canvas* showing the new lens field dimensions.

Tip: To use an existing mark file with a new lens having a different field size, follow steps 1 – 5 above and then open your existing mark file. Select all objects (from the *Edit* menu, choose Select All) and paste them into your new, blank drawing. All selected objects and their corresponding object properties are pasted into the new drawing. Save this new drawing with another name or append the lens size to the old filename.

Setting measurement units

You can set WinMark Pro to display all measurements in inches, centimeters, or millimeters.

To change measurement units, perform the following steps:

1. From the *Tools* menu, select General Settings….
2. In the *Applications Settings* dialog box, click the *Application Settings* tab.
3. Click Show Units Type (Figure 3-2), and then click the arrow.
4. From the drop-down list, select the appropriate measurement units.
5. Click *Apply* to apply the change, and then click *OK*.

Note: When using WinMark Pro's ActiveMark (ActiveX) controls, all units of measure are calculated in inches. Your application or user-interface must convert measurement values from centimeters or millimeters to inches as required. Make sure that the Show Units Type property is set to inches—unpredictable object placement or operation may result if the Show Units Type is set to millimeters or centimeters.

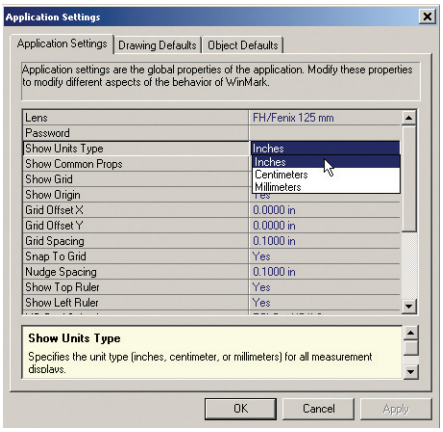


Figure 3-2 Changing "Show Units Type"

4 WinMark Pro Drawing Editor

The *WinMark Pro Drawing Editor* chapter includes subsections:

- Main Window
- Toolbar
- Tool Box
- Drawing Canvas
- Slide Bar
- Object List
- Property List
- Help Window
- Status Bar

Main Window

To aid creation of mark files, the Drawing Editor's Main Window (Figure 4-1) is divided into several distinct areas. Each item is described in further detail below.

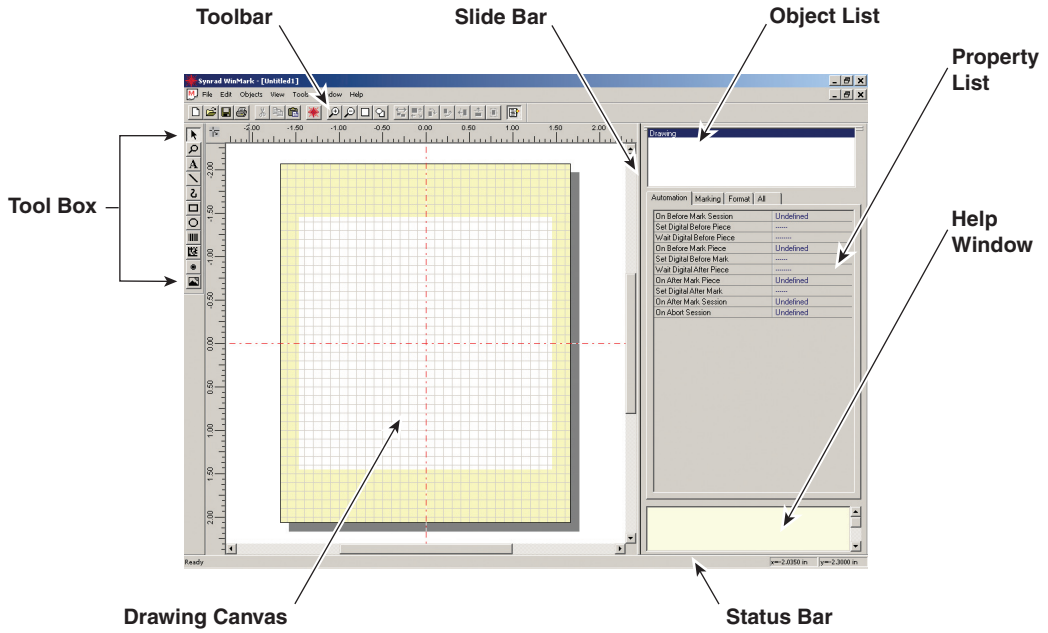


Figure 4-1 WinMark Pro Main Window

Toolbar

Icons on WinMark Pro's *Toolbar* (Figure 4-2) provide shortcuts to common editing functions. To use a shortcut, select one of more objects on the *Drawing Canvas* and click the appropriate *Toolbar* button.

Figure 4-2 Toolbar icons

The following shortcuts are provided on the Toolbar:

New – create a new, blank drawing. *Drawing Canvas* size is based on the current Lens selection.

Open – open an existing WinMark Pro (.mkh) mark file.

Save – save the current active mark file.

Print Drawing – print the active drawing.

Cut – cut selected objects and place them on the Clipboard.

Copy – copy selected objects to the Clipboard.

Paste – paste the current Clipboard contents into the drawing.

Mark – open the *Launcher* window in preparation for marking. Once the *Launcher* window is open, press F1 to initiate marking. *Launcher* remains open until closed by the user.

Zoom In – Zoom in towards the center of the current view.

Zoom Out – Zoom out away from the center of the current view.

Zoom Page – Zooms the view to fit the *Drawing Canvas* in the *Main Window*.

Zoom Extents – Zooms the view to encompass all mark objects in the drawing, even those objects placed off the *Drawing Canvas*.

Align – opens a dialog that allows you to align two or more objects horizontally and/or vertically.

Transform – opens a dialog box that allows you to precisely resize, scale, rotate, or position objects.

Rotate +45 – rotate selected objects clockwise in 45 degree increments. Note that bitmap objects rotate only in 90° increments.

Rotate –45 – rotate selected objects counterclockwise in 45 degree increments. Note that bitmap objects rotate only in 90° increments.

Mirror X – mirror selected objects in the X (horizontal) axis.

Mirror Y – mirror selected objects in the Y (vertical) axis.

Center – center selected objects around the (0, 0) origin point.

Toggle Property Window – toggle the *Property Window* display On or Off.

Tool Box

Icons in the *Tool Box* (Figure 4-3) represent the drawing and editing tools you use in WinMark Pro. To select a tool, place the cursor over the tool and click the left mouse button. The cursor changes into the appropriate icon. When drawing with a *Line*, *Freehand*, *Rectangle*, or *Arc Tool*, press and hold the left mouse button while dragging to create the desired size, length, or direction and then release the mouse button.

Figure 4-3 Tool Box

The following tools are available in the Tool Box:

Selection Tool – selects or deselects objects on the *Drawing Canvas*. Click the left mouse button to select an object. Hold the left mouse button and drag to select multiple objects within the selection area, or hold the CTRL key and left click objects to add them to the selection set. Click on a blank section of the drawing to deselect all objects. Click the *Selection Tool* icon to regain the arrow cursor after using a tool such as the *Zoom Tool*, which does not default to an arrow cursor.

Zoom Tool – zooms in or out of the *Drawing Canvas*. After selecting the *Zoom Tool*, click the left mouse button to zoom in; click the right mouse button to zoom out. The "zoomed" image is centered on the position of the *Zoom Tool* when you click. To enlarge a specific area, hold the left mouse button and drag to define the zoom area.

Tip: The following tools add or import objects onto the *Drawing Canvas* using WinMark Pro's default set of object properties. You can edit these properties (on the *Tools/General Settings…/Object Defaults* tab) so that any new objects created reflect the specific settings required for your application.

Text Tool – places a text object in the drawing. Click the *Text Tool* on the *Drawing Canvas* to set an insertion point and open the *Text Caption Editor* dialog box. After entering text in the dialog box, you can select the text box to modify its position or change text attributes such as font type or size using text properties shown in the *Property List*.

Line Tool – draws straight point-to-point lines. If Snap to Grid (*Tools/General Settings…/Application Settings* tab) is *Yes*, the line's origin and destination endpoints snap to the nearest grid line intersection.

Freehand Tool – draws irregular, freehand lines. If Snap to Grid is *Yes*, any curves drawn are snapped to the nearest grid line and intersection.

Rectangle Tool – draws square or rectangular objects. If Snap to Grid is *Yes*, lines and corners are snapped to the nearest grid line and intersection.

Arc Tool – draws elliptical and circular objects. By default, the *Arc Tool* draws ellipses, hold the SHIFT key down while dragging to constrain the object to a circle. If Snap to Grid is *Yes*, the arc's outer diameter is snapped to the nearest grid line.

Bar Code Tool – places a bar code object in the drawing. Click the *Bar Code Tool* then click in the drawing to set an insertion point. Barcode Number is highlighted on the *Format* tab. Type the characters you want converted into bar code form. Modify bar code attributes on the *Format* tab.

2D Code Tool – places a 2D Data Matrix or QR Code object in the drawing. Click the *2D Code Tool* then click in the drawing to set an insertion point. 2D Barcode Text is highlighted on the *Format* tab. Type the characters you want converted into 2D code form. Modify 2D code attributes on the *Format* tab.

Spot Tool – places spots on the drawing that represent the diameter of the laser beam. Spots are useful as guides for aligning the laser beam. If the correct Power and Spot Lase Duration values are specified in the *Property List*, spots can be used to perforate, or drill, the workpiece.

Graphic Tool – imports graphic image files into the drawing. Click the *Graphic Tool* to open the *Open* dialog box. Select a file type, click on the file to import, and then click *Open* to place the graphic on the *Drawing Canvas*.

Drawing Canvas

The *Drawing Canvas* is the area where graphics and text objects are arranged to create a mark file for laser marking the workpiece. The white center area on the *Drawing Canvas*, shown in Figure 4-4, is the optimal mark area for the currently installed lens. The yellow shaded border indicates the maximum marking field of the lens. Note that objects placed in the shaded area, especially near the corners, may exhibit a slight degradation in mark quality.

Overall dimensions of the *Drawing Canvas* correspond to the focusing lens size entered in the *Lens Selection* dialog box. These dimensions, Field Width and Field Height, should not be altered as they affect the overall scaling and proportionality of the marked image.

Dotted *Origin Lines* running across the canvas identify the horizontal and vertical centers of the work area. The intersection of the *Origin Lines* (X=0, Y=0) indicates the center of the focusing lens field and provides a reference for aligning the image on the parts to be marked. Figure 4-4 shows X-Y values for the four quadrants defined by the *Origin Lines*. Note that WinMark Pro's XY coordinates are not Cartesian coordinates—positive X values are located to the right of the vertical origin while positive Y values are located below the horizontal origin.

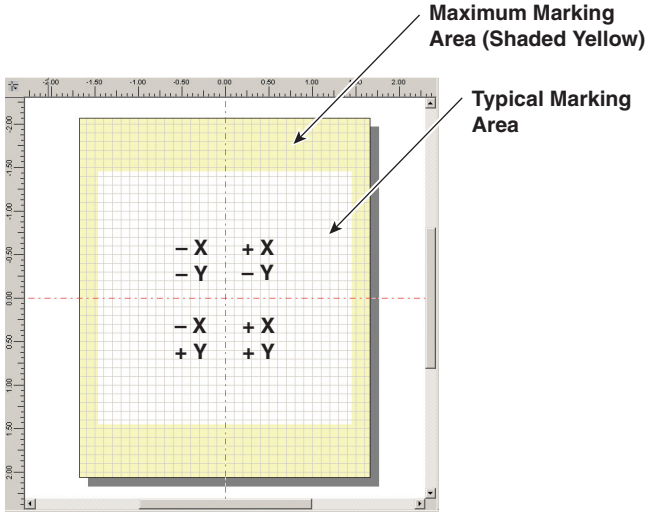


Figure 4-4 Drawing Canvas

Slide Bar

Grab the *Slide Bar* with the Pointer and pull to the left or right to change the horizontal display area of the *Drawing Canvas* and Object/Property windows.

Object List

Each object placed on the *Drawing Canvas* is added to the *Object List* and is associated with a set of object properties that define some characteristic of that object. For example, drawing a single rectangle creates an entry in the *Object List* named "*Polyline1*". The *Drawing Canvas* itself also contains a unique set of properties and always appears first in the *Object List* as "*Drawing*".

Property List

The *Property List* consists of six tabs: *Special*, *Automation*, *Tracking*, *Marking*, *Format*, and *All*. These tabs group sets of properties by function. When an entity on the *Object List* is selected, properties that affect that entity are displayed on the appropriate tabs in the *Property List* window. Those individual properties can then be viewed or edited.

Note: The *Tracking* tab only appears when WinMark Pro is actively communicating with an FH Series Tracker marking head or a Fenix Tracker Laser Marker.

Help window

When a property is selected from a *Property List* tab, a description or explanation of that property appears in the *Help* window.

Status Bar

The *Status Bar* displays information about tools and their current XY position, as well as information about icons on the toolbar.

Note: Use the *Toggle Property Window* toolbar button ( ) to show/hide *Object List*, *Property List*, and *Help* window views.

5 Creating a drawing

The *Creating a drawing* chapter includes subsections:

- Opening a new drawing
- Adding text
- Adding bar codes
- Adding other objects
- Importing a graphics file
- Precision positioning

Opening a new drawing

WinMark Pro creates a new drawing with a blank *Drawing Canvas* each time it opens. If WinMark Pro is already open, select *New* from the *File* menu.

Adding text

1 Select the *Text Tool* (A) from the *Tool Box*.

2 Position the cursor on the *Drawing Canvas* and click the left mouse button to set an insertion point. The *Text Caption Editor* dialog box appears.

3 In the *Text Caption Editor* dialog box (Figure 5-1), enter the desired text string. If multi-line text is required, press ENTER to start a new line of text.

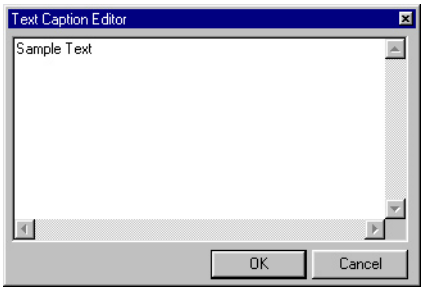


Figure 5-1 Text Caption Editor dialog box

4 Click *OK*. The text appears on the canvas within a bounding outline.

Use the Font Type property to select either stroke or TrueType® fonts. Any TrueType font installed in Window's *Fonts* folder is available for laser marking. Experiment with changing Font Type, Text Justification, and other text properties on the *Format* tab.

Adding bar codes

1D bar codes

1 Select the *Bar Code Tool* ( ) from the *Tool Box*.

2 Position the cursor on the *Drawing Canvas* and click the left mouse button to set an insertion point.

3 In the *Property List*, click the *Format* tab and enter information in the Barcode Number field.

4 Select the appropriate Barcode Style for the code type you wish to mark. Experiment with changing various 1D bar code properties on the *Format* tab.

2D codes

1 Select the *2D Code Tool* ( ) from the *Tool Box*.

2 Position the cursor on the *Drawing Canvas* and click the left mouse button to set an insertion point.

3 In the *Property List*, click the *Format* tab and then enter data in the 2D Barcode Text field or click the ellipsis (…) to enter data in the *Text Caption Editor* dialog box.

4 Select the appropriate 2D Barcode Style for the 2D code type you wish to mark. Experiment with changing various 2D code properties on the *Format* tab.

Adding other objects

Rectangles and arcs

1 Select the *Rectangle Tool* ( ) or *Arc Tool* ( ) from the *Tool Box*.

2 Click and hold the left mouse button to set a starting point, and then drag to define the object's size and shape. Hold the SHIFT key down while dragging the mouse to create squares instead of rectangles (*Rectangle Tool*) or circles instead of ellipses (*Arc Tool*).

3 Release the mouse button when the desired size and shape is reached.

Tip: To precisely size objects, open the *Object Transformations* dialog, uncheck *Maintain Aspect Ratio*, and type in specific values (to four-digit accuracy!) in both *New Width* and *New Height* fields.

Spots

1 Select the *Spot Tool* ( ) from the *Tool Box*.

2 Position the cursor on the *Drawing Canvas* and click the left mouse button to set an insertion point.

3 In the *Property List*, click the *Format* tab and enter a Spot Lase Duration time in milliseconds (ms).

Importing a graphics file

1 Select the *Graphic Tool* ( ) from the *Tool Box* or, on the *File* menu, click *Import*. The *Open* dialog box appears, asking you to locate a file for import. For demonstration purposes, we will import a sample vector graphics file included with WinMark Pro.

Note: The File Import Filters (automatically installed by WinMark Pro versions 4.0.0.4018 and above) are required to import vector graphic files. When running older WinMark versions, remember to install File Import Filters from the WinMark Pro CD or download and install them from the WinMark Pro web site.

2 In the *WinMark* folder, double-click the *Samples* folder.

3 In the *Samples* folder, double-click the *SampleImports* folder.

4 In the *Files of type* drop-down list, select (*.eps) Encapsulated PostScript.

5 Select the *Synrad.eps* file and click *Open*. The SYNRAD logo appears on the *Drawing Canvas* (Figure 5-2).

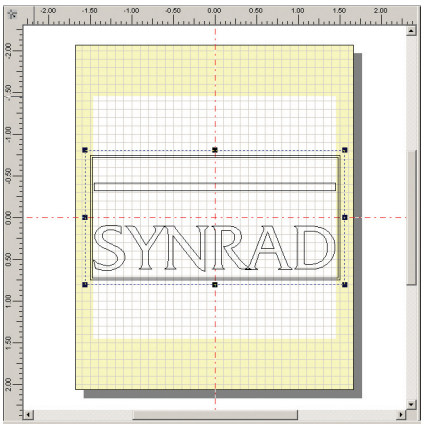


Figure 5-2 Sample imported graphic file

Tip: Control scaling of imported vector files using the Vector Import Scale property (*Tools/General Settings…/Object Defaults* tab). The default setting is "One to one", which means that vector drawings import at full size. Depending on the lens field/*Drawing Canvas* size, the image may extend out of the marking field. If this occurs, change scaling to "Best Fit", "X axis", or "Y axis" and re-import the image. You can also grab a corner handle of the bounding box with the *Selection Tool* and resize the image proportionally by dragging the pointer. Note that the marking head will not mark any portions of objects that extend off the *Drawing Canvas*.

Precision positioning

In many applications, very precise dimensions and mark placement are required when laser marking objects. In these cases, use the *Object Transformations* dialog box to precisely resize, scale, or position selected objects on the *Drawing Canvas*.

1 Add an object to the *Drawing Canvas* or use the *Selection Tool* and click to select an existing object in the *Drawing*.

Note: When an object is selected, a bounding outline appears around the selected object or objects.

2 From the *Objects* menu, click Transformations… or click the *Transform* button ( ) on the *Menu Bar*.

3 In the *Object Transformations* dialog box (Figure 5-3), change X- and Y-position values (with four-digit accuracy!), scale the selection by entering a percentage, or resize by entering precise width and height dimensions.

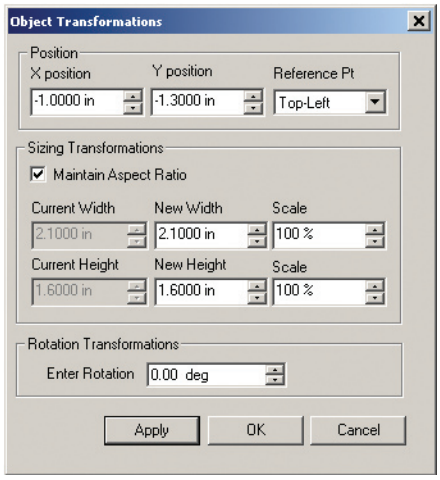


Figure 5-3 Object Transformations dialog box

4 Click *Apply* to preview transformed objects without closing the *Object Transformations* dialog box. Click *OK* to accept the transformation and close the dialog or click *Cancel* to leave object position unchanged.

6 File development

The *File development* chapter includes subsections:

- Off-line mark file development
- FH Smart mark file development

Off-line mark file development

Off-line mark file development is possible on any Windows computer running WinMark Pro. Without a *Hardlock* or FLCC card installed, WinMark Pro reverts to "demo" mode allowing you to create and save files. The only missing functionality is the ability to send microvector (mark) data to the marking head. You can always download the latest version of WinMark Pro free of charge from the WinMark Pro web site at http://winmark.com or from the WinMark Pro CD.

In order for .mkh files to open and mark, remember that the build number (×.×.×.***1234***) of the off-line development version of WinMark must be less than or equal to the build number of the WinMark Pro version installed on the marking computer.

Note: Because Tracking parameters are specific to each individual Tracker application, properties on the *Tracking* tab cannot be viewed or modified unless WinMark Pro is actively communicating with a Tracker-capable FH Series or Fenix marking head.

FH Smart mark file development

FH Smart mark files are developed using WinMark Pro in "demo" mode. Because FH Smart executes mark files in firmware, any version of WinMark Pro can be used for mark file development. Text automation features such as serialization, date coding, time stamping, etc. are setup in the same manner as a normal Index or Tracker file. With the FH Smart head however, these functions are processed by the marking head in real-time and no computer interface is required.

Mark files are downloaded to FH Smart through an RS-232 serial interface using SmartTerm, a sample terminal-emulation program or by using WinMark Pro v5.0. Alternatively, the *FH Series Smart Operator's Manual* describes the FH Smart communications protocol in detail for those customers interested in developing their own custom user interface.

7 WinMark Launcher

The *WinMark Launcher* chapter includes subsections:

- Introduction
- Launcher display
- Opening a drawing
- Marking parts
- Modifying a drawing

Introduction

WinMark Launcher allows you to open and mark existing mark files outside of the Drawing Editor environment. This eliminates the possibility of inadvertent changes to archival mark files and, as shown in Figure 7-1, provides a simplified graphical interface for operators.

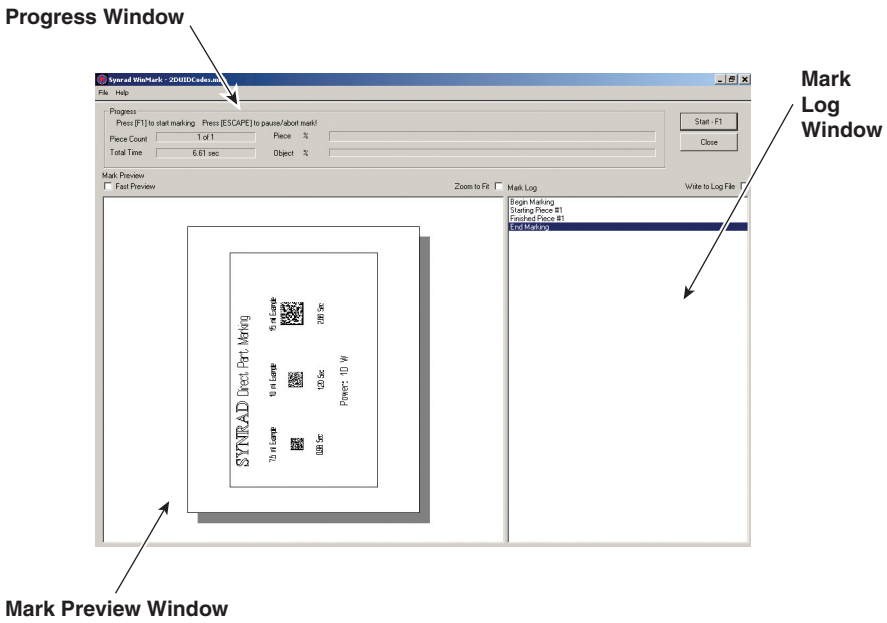


Figure 7-1 Launcher window

Start WinMark Launcher by double-clicking the Launcher icon ( ) on your desktop, or perform the following steps:

1 Ensure that other WinMark programs including WinMark Pro and Digital Scope are closed.

Note: After installing or upgrading to a new version, open WinMark Pro first, before opening Launcher. This allows WinMark Pro to initialize lens selection and FLCC data in the computer's *System Registry* before a mark session begins.

2 Click *Start* on the taskbar.

3 Select *Programs* and then locate the *Synrad WinMark* folder.

4 Click the *Synrad WinMark Launcher* icon.

5 Launcher opens and automatically loads the last file marked.

Launcher display

Launcher's graphical display comprises several distinct windows as shown in Figure 7-1. Each is described below.

Progress window

The Progress window provides status information about the ongoing mark session. *Piece Count 1 of X* provides a running total of pieces marked in the current mark session in cases where *Mark Count* is zero or greater than one. *Total Time* displays the total elapsed time in this mark session. *Piece %* is a progress bar illustrating percent of marking complete on the current piece (part to be marked). *Object %* illustrates the percent of marking complete on the current object in the mark file.

Mark Preview window

Mark Preview displays the actual mark being made in real-time. During marking, a red outline box surrounds the object currently being laser marked.

Fast Preview check box

Checking *Fast Preview* grays out individual elements in the mark file, showing only mark location. In time-critical applications, this option saves the small amount of time involved in redrawing the screen.

Zoom to Fit check box

Selecting the *Zoom to Fit* check box zooms the Launcher view to the extents of the mark. This view is useful when an operator must verify a manual text input, serial port read, or serialization event while marking very small objects.

Mark Log window

The Mark Log records and displays every step or event in the mark sequence as it occurs in real-time during the marking process.

Write to Log File check box

Checking *Write to Log File* creates a Windows Log File (.log) and writes each log event displayed in the Mark Log window to the marking computer's hard drive. Selecting the *Write to Log File* check box opens the *Save As* dialog box which prompts you for a filename and storage location. Using the *Write to Log File* feature will affect cycle times of the mark because each event is written to the hard drive in real-time.

Data in the saved mark log is formatted as follows:

DD MM HH:MM:SS "mark event description"

where DD is the two-digit day, MM is the two-digit month, and HH:MM:SS are Hours:Minutes:Seconds as read from the system clock. "Mark event description" is a copy of the same data displayed in the *Mark Log* window.

Opening a drawing

To open an existing mark file, perform the following steps:

1. Click Open Drawing… under the *File* menu.
2. In the *Open* dialog box, locate and select the file you wish to mark.
3. Double-click the file or click *OK* to load the selected mark file.

Marking parts

Once a mark file is loaded into Launcher, begin marking by pressing F1 or click the *Start-F1* button. By changing *Startup Option* settings in WinMark Pro, Launcher can be configured to (1) load a specific file when opened, (2) automatically launch when the computer boots-up, and (3) automatically start marking when Launcher opens.

Note: To prevent I/O conflicts, always ensure that WinMark Pro and Digital Scope are closed before attempting to operate the marking head using the membrane panel pushbuttons or WinMark Launcher.

⚠ Danger
serious personal injury

When the *Start marking immediately after Launcher loads* check box is selected, the laser will begin firing immediately after WinMark Launcher loads on start-up, provided that all input states set in the mark file's Event Automation, if any, are valid.

Because exposure to 10.6 µm laser radiation can inflict severe corneal injuries and seriously burn human tissue, appropriate safeguards must be in place to prevent serious personal injury or equipment damage.

Tip: When running time-sensitive applications, minimizing the Launcher window helps to shorten cycle times by reducing the time that the Windows Operating System spends on refreshing the display.

Modifying a drawing

To edit the currently open mark file, perform the following steps:

1. Click Modify Drawing… under the *File* menu.
2. If a user password has been set, then the *Enter Password* dialog box opens. Type the correct case-sensitive password and click *OK*.
3. Edit the mark file as required in WinMark's *Drawing Editor*.
4. When editing is complete, save the file and close WinMark Pro before continuing to mark from Launcher.

8 WinMark Digital Scope

Digital Scope is a stand-alone program designed to test or verify the functionality of WinMark Pro's laser control and I/O capability through the Fiber Link Controller Card (FLCC) and the digital fiber optic cable.

Start Digital Scope by double-clicking the *Digital Scope* icon ( ) on your desktop, or perform the following steps:

1 Ensure that other WinMark programs including WinMark Pro and WinMark Launcher are closed.

Note: After installing a new or upgraded version, WinMark Pro must be opened first, prior to opening Digital Scope. This enables lens selection and FLCC data to be initialized before a mark session begins.

2 Click *Start* on the taskbar.

3 Select *Programs* and then locate the *Synrad WinMark* folder.

4 Click the *Synrad Digital Scope* icon.

5 Digital Scope opens as shown in Figure 8-1.

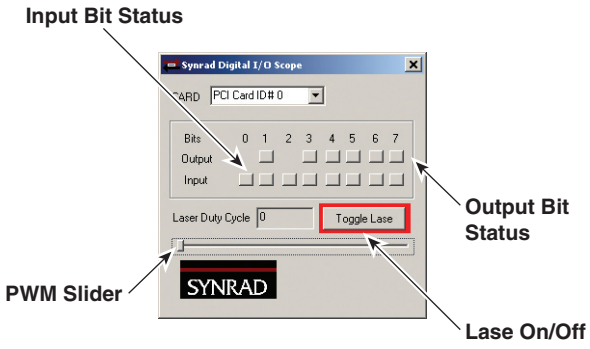


Figure 8-1 Digital Scope display

A description of each Digital Scope control follows:

Input Bit Status

Input bit "buttons" pop-in to indicate that an input connected to the marking head or Fiber Link Controller Card (FLCC) is active. See Chapter 12, *Automation basics*, for a list of valid inputs for your particular marking head.

PWM Slider

Click and drag the PWM slider to set output laser power by varying the laser's pulse width modulation (PWM) between 0–100% duty cycle. Power output is displayed in the *Laser Duty Cycle* display box.

Output Bit Status

Click an output bit "button" to activate an output connected to the marking head or FLCC. See Chapter 12, *Automation basics*, for a list of valid outputs for your particular marking head.

Lase On/Off

Click the *Toggle Lase* button to turn the laser On. The laser will fire at the specified *Laser Duty Cycle*. Click again to toggle the laser Off.

Note: To prevent I/O conflicts, always close Digital Scope before attempting to operate the marking head using the membrane panel pushbuttons or before starting WinMark Pro or WinMark Launcher.

Note: Beginning with WinMark Pro version 4.0.0.3773, Digital Scope now sends a command to center FH Series marking head/Fenix Laser Marker galvanometers when the *Toggle Lase* button is pressed.

Important Note: If the Fast Acting Safety Interlock (FASI) feature is enabled on the FLCC, then *Input Bit 3* (*IN3*) on FH Series or Fenix Laser Markers, or *Input Bit 5* on DH Series systems, must be active (high level voltage applied) before the laser will fire.

Basic Operation

Use information in this section to begin laser marking.

This section contains the following chapters:

- Chapter 9 – Marking basics
  explains how font selections and graphic file formats affect mark speed and quality.
- Chapter 10 – Marking parts
  explains how to mark single pieces or parts in batches.
- Chapter 11 – Fast (spot-style) 2D codes
  demonstrates how to create fast marking, vector-style 2D codes.
- Chapter 12 – Automation basics
  describes the basic procedures in connecting to automated parts handling equipment.
- Chapter 13 – ActiveMark automation
  describes the basics of WinMark Pro's ActiveMark (ActiveX) technology when used in a Microsoft Visual C++ and MS Visual Basic programming environment.
- Chapter 14 – Custom Test Mark feature
  explains the FH Series/Fenix custom test mark feature.
- Chapter 15 – Tracker marking
  contains information on determining approximate Tracking line speeds and illustrates marking of text and graphics objects.

9 Marking basics

The *Marking basics* chapter includes subsections:

- Types of marks
- Marking parameters
- Optimization

When creating a mark file, your choice of font for text and your choice of file format for imported graphic images are the primary factors that influence the overall speed and quality of the mark. The following sections lead you through the creation of text and graphic files to illustrate how your choices affect marking. The *Optimization* section provides examples of how to optimize a file for speed or for mark quality.

⚠ Danger
serious personal injury

Before performing the marking tutorials in this chapter, ensure that all personnel in the area are wearing protective eyewear. Read and follow all laser safety and fume extraction precautions described in your laser and marking head Operator's Manuals.

CO_2 lasers emit ***invisible*** infrared laser radiation at the 10.6 µm CO_2 wavelength. Since direct or diffuse laser radiation can inflict severe corneal injuries, always wear eye protection when in the same area as an exposed laser beam. Do not allow the beam to contact a person. CO_2 lasers emit an invisible beam that is capable of seriously burning human tissue.

Always be aware of the beam's path and always use a beam block while testing.

⚠ Warning
serious personal injury

In order to halt marking before a mark has been completed, you must press the ESC key on the keyboard. In some instances, you must press and hold the ESC key until marking is halted. Because WinMark Pro is busy sending microvector and control data to the marking head in real-time, keyboard status is checked only intermittently during a mark.

To halt lasing immediately you can toggle the laser's *Keyswitch* to Off, open the laser's *Remote Reset* input, or open the FASI input (if the Fast Acting Safety Interlock is enabled). Although lasing is disabled immediately, WinMark Pro is unaware of laser status and will continue the mark session until it is complete or halted.

Important Note: Power settings described in this tutorial section are based on the use of a 25 W laser marking an anodized aluminum substrate with a 125 mm focusing lens. When working through the examples below, be sure to adjust power levels accordingly for higher or lower power lasers, for the spot size (power density) of your lens, and for the specific substrate you are marking.

Types of marks

There are two distinct methods used to mark objects. The first method, vector marking, is made by moving the beam from one X-Y point to another X-Y point (a vector) along a straight or curved path. This type of mark is very fast, limited only by the speed of the galvanometers moving the beam-steering mirrors. The second method of marking, raster marking, is accomplished by switching the beam on and off as it scans across the image area. At the end of a scan line, the beam returns, is moved down one step, and then makes another pass. This rastering continues through the length of the image. If bi-directional rastering is chosen, marking is enabled as the beam moves both directions across the image. In addition, you can specify either a horizontal or a vertical orientation for rastering. Raster marking is slower than vector marking, but rastering can create filled objects and text or even reproduce photographic images. Figure 9-1 illustrates both types of marking.

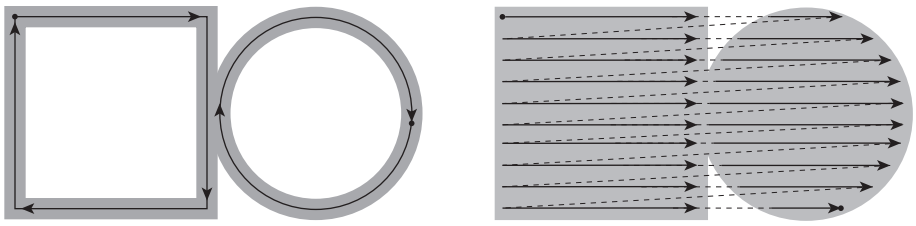


Figure 9-1 Vector versus raster marking

Text

When creating text, there are two categories of fonts to choose from: (1) stroke fonts or (2) TrueType® fonts. Stroke fonts are the type of fonts typically used in CAD programs because they are drawn (or marked) very quickly using point-to-point vectors. Twelve different stroke font styles are built into WinMark Pro® to improve high-speed coding of text, date codes, and serial numbers.

Any TrueType font installed in Windows's *Fonts* folder is available for use by WinMark Pro. If Text Fill is set to *No*, the font is marked as a vector outline; if Text Fill is *Yes*, the font is filled by rastering the beam, effectively "coloring in" the outline to produce a solid fill. Figure 9-2 illustrates examples of stroke, TrueType, and filled TrueType fonts.

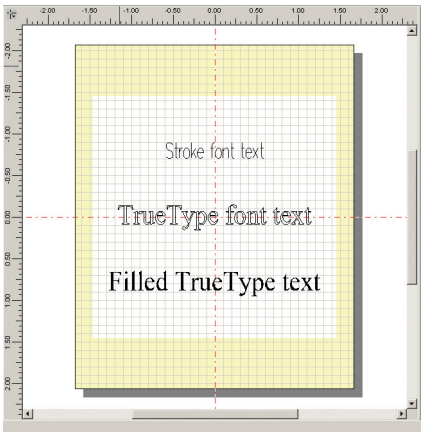


Figure 9-2 Font types

The tutorial section below illustrates the relative marking speeds of stroke and TrueType fonts.

Mark a stroke font:

1. Select the *Text Tool* from the *Tool Box*, position it on the *Drawing Canvas*, and click the left mouse button to set an insertion point. The *Text Caption Editor* dialog box appears.
2. In the *Text Caption Editor* dialog box, type: "ABCDEFG123".
3. Click *OK*. ABCDEFG123 appears on the canvas within a bounding outline.
4. On the *Format* tab, set Font Type to *Stroke* and then in Stroke Font, select a stroke font to mark.
5. Choose a Text Height that will allow the text to easily fit on the *Drawing Canvas*.
6. Click the *Marking* tab (Figure 9-3) and set Velocity to *30* in./sec., Power to *60%*, and Resolution to *600*.

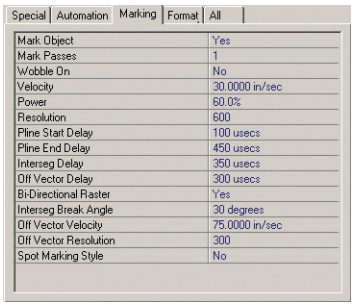


Figure 9-3 Marking tab object properties

7 Position a sheet of anodized aluminum under the lens, ensure all personnel are wearing safety glasses, and then click the *Mark* button.

8 When the *Synrad Marking Progress* dialog box appears, click *Start-F1*.

9 Observe how the marking head is producing the mark. After completion of the mark, note the cycle time.

10 Save the file as *Stroke demo.mkh*.

Mark an unfilled TrueType font:

1 Open a new drawing.

2 Select the *Text Tool*, position it on the *Drawing Canvas*, and click the left mouse button to set an insertion point.

3 In the *Text Caption Editor*, type: "ABCDEFG123".

4 On the *Format* tab, set Font Type to *TrueType* and then in TrueType Font, select a font to mark. Set Text Filled to *No* and set Text Curve Detail to *600*.

5 Set Text Height to the same value used in the stroke font example.

6 Click the *Marking* tab and set Velocity to *30* in./sec., Power to *60%*, and Resolution to *600*.

7 Position a sheet of anodized aluminum under the lens, ensure all personnel are wearing safety glasses, and then click the *Mark* button.

8 When the *Synrad Marking Progress* dialog box appears, click *Start-F1*.

9 Observe how the marking head is producing the mark. After completion of the mark, note the cycle time.

10 Save the file as *Unfilled TrueType demo.mkh*.

Mark a filled TrueType font:

1 Open a new drawing.

2 Select the *Text Tool* and click to set an insertion point.

3 In the *Text Caption Editor*, type: "ABCDEFG123".

4 On the *Format* tab, set Font Type to *TrueType* and then in TrueType Font select the same font used in the unfilled font example above.

5 Set Text Filled to *Yes*, Text Curve Detail to *600*, and set Text Height to the same value used in the stroke and unfilled font examples.

6 Click the *Marking* tab and set Velocity to *40* in./sec., Power to *50%*, and Resolution to *600*.

7 Position a sheet of anodized aluminum under the lens, then mark the text.

8 Again, observe how the marking head is producing the mark and note the cycle time.

9 Save the file as *Filled TrueType demo.mkh*.

The three text examples just demonstrated show the importance of choosing fonts to match the marking application. In processes where speed is an issue such as in marking part numbers, date codes, or other identifying information at high throughput speeds, stroke (vector) fonts can mark one and one-half times faster than unfilled (vector) TrueType fonts and ten times faster than filled (raster) TrueType fonts. When using filled TrueType fonts to mark products that are highly visible or that must convey a quality "look" or match a corporate logotype, longer cycle times are inevitable.

Graphics

When creating graphics, there are two basic types of file formats: (1) bitmap and (2) vector graphics. Bitmap (raster files) such as TIFF, BMP, and JPEG formats are the types of files created by paint programs. Bitmap file elements are marked using a raster scanning technique. Most CAD and illustration programs generate vector files in formats such as EPS and DXF formats. Vector file elements are marked quickly using point-to-point vector movements. A third graphics format, metafile, can contain both bitmap and vector elements. Common metafile formats include WMF and CGM.

Mark a bitmap graphics file:

1 Open a new drawing and then click the *File Import Tool*. The *Open* dialog box appears, asking you to locate a file for import. For illustration purposes, we will import a sample file included with WinMark Pro.

2 In the *WinMark* folder, double-click the *Samples* folder.

3 In the *Samples* folder, double-click the *SampleImports* folder.

4 In the *Files of type* drop-down list, select (*.tif) Tagged Image File Format.

5 Select the *Family.tif* file and click *Open*. The bitmap file is shown in Figure 9-4.

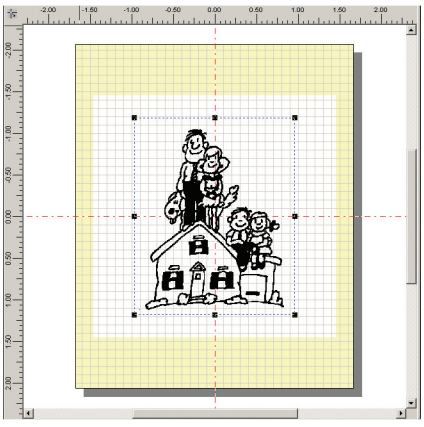


Figure 9-4 Family.tif

6 Click on the family image to select it, then use the *Object Transformations* dialog box to scale the image to fit the *Drawing Canvas*.

7 Click the *Marking* tab and set Velocity to 40 in./sec., Power to 60%, and Resolution to 300.

8 Position a sheet of anodized aluminum under the lens, ensure all personnel are wearing safety glasses, and then click the *Mark* button.

9 When the *Synrad Marking Progress* dialog box appears, click *Start-F1*.

10 Observe how the marking head is producing the mark. After completion of the mark, note the cycle time.

11 Save the file as *Bitmap demo.mkh*.

Mark a vector graphics file:

1 Open a new drawing and then click the *File Import Tool*. The *Open* dialog box appears, asking you to locate a file for import. For illustration purposes, we will import a sample file included with WinMark Pro.

2 In the *WinMark* folder, double-click the *Samples* folder.

3 In the *Samples* folder, double-click the *SampleImports* folder.

4 In the *Files of type* drop-down list, select (*.dxf) AutoCAD Drawing Interchange.

5 Select the *Chairbin.dxf* file and click *Open*. *Chairbin.dxf* (Figure 9-5) is a vector graphic illustrating an office chair as a wire-frame line drawing.

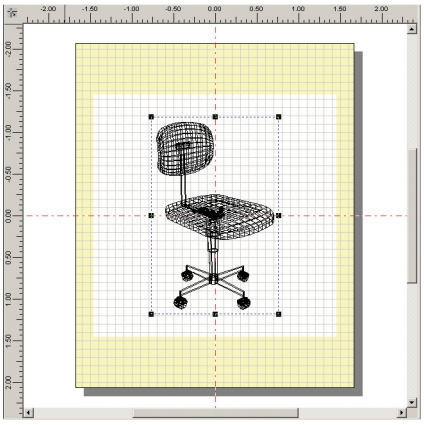


Figure 9-5 Chairbin.dxf

6 Click on the Chairbin image to select it.

7 Click the *Marking* tab and set Velocity to 60 in./sec., Power to 80%, and Resolution to 300.

8 Position a sheet of anodized aluminum under the lens, then mark the drawing.

9 Observe how the marking head is producing the mark. After completion of the mark, note the cycle time.

10 Save the file as *Vector demo.mkh*.

Vector graphics are used for marking objects at high throughput speeds when cycle times are important. Bitmap graphics are appropriate when a filled logo or photographic image is being marked and quality is the most important marking factor. You can import both vector and bitmap files into the same drawing to shorten cycle times and obtain the best quality mark possible.

Bar codes

1 Open a new drawing.

2 Select the *Bar Code Tool* and click to set an insertion point.

3 On the *Format* tab, in the Barcode Number field, type: "123456".

4 On the *Format* tab, set Barcode Style to *Code 3 of 9*, Barcode Resolution to *200*, and Barcode Show Text to *No*.

5 If coding a dark material that will mark lighter, set Barcode Inverted to *Yes*.

Tip: Most bar codes are marked, or printed, as dark lines on a light background because the bar code must create a sharp color contrast against the material's background in order for the bar code to be readable. When creating a contrasting mark on dark material, set Barcode Inverted to *Yes*. This causes WinMark to mark the spaces between bars (instead of the bars) and add a "quiet zone" around the code so that a contrasting space between dark bars is created.

---

6 Click the *Marking* tab and set Velocity to *30* in./sec., Power to *60*%, and Resolution to *300*.

Note: Setting the Resolution value too high may introduce reader errors if line width begins to encroach on the white space between bars.

7 Position a sheet of anodized aluminum under the lens, then mark the bar code.

Tip: When marking plastics (that contrast) or ink-coated surfaces, start with Barcode Resolution settings in the range of 200–250. Bar codes on mild and stainless steels mark best at resolutions around 600–650. To decrease mark (cycle) time, Barcode Resolution should not be set much higher than the lowest resolution that provides consistently readable results.

2D codes

1 Open a new drawing.

2 Select the *2D Code Tool* and click to set an insertion point.

3 On the *Format* tab, in the 2D Barcode Text field, click the ellipsis (…). The *Text Caption Editor* dialog box appears.

4 In the *Text Caption Editor* dialog box, type: "ABCDEFG123" and click *OK*.

5 In the 2D Barcode Style field, select either *Data Matrix* or *QR Code*.

6 Set Barcode Inverted to *Yes* if marking a dark material.

7 Click the *Marking* tab and set Velocity to *30* in./sec., Power to *60%*, and Resolution to *300*.

Note: Setting the Resolution value too high may lead to cell growth errors if filled cells begin to encroach on the space allocated for unfilled cells.

8 Position a sheet of anodized aluminum under the lens, then mark the code.

Tip: For best results when marking 2D codes, start with Resolution settings in the range of 600–750. Use higher resolutions when marking low contrast materials to ensure that all information is properly read.

Marking parameters

Once a decision has been made to use either a stroke or TrueType font or to use either a bitmap or vector graphic, the three most important object properties are: (1) Velocity, (2) Power, and (3) Resolution. These three parameters plus object delays on the *Marking* tab are the primary factors that determine the speed and quality of the mark.

Velocity

Marking velocity (X-Y galvanometer speed) determines how fast each microvector is marked. Setting galvanometer velocity too high affects mark quality as delay settings become much more critical when velocity increases. Velocity also affects power since a faster moving beam has less heat effect on the material. Velocity is adjustable in a range of 0.01 to 400 in./sec. (0.25–10,160 mm/sec.).

Power

Laser output power must be optimized for the material being processed and for the type of mark required. Power can be set from 0 to 100% of the laser's actual power output. Typically, 70 to 80% will equal the laser's specified output power in watts. Applying too much power may engrave too deeply or char the material unacceptably.

Tip: When marking certain plastics, such as PEEK, PVC, brown Bakelite and dark phenolic, contrasting marks are often achieved using very low power settings (1 to 2 watts).

Resolution

Resolution, actually microvectors per inch, is a primary factor in mark quality. Increasing resolution increases the quality, or "crispness" of the image. Reducing resolution too much can make marks look "rough". Practical resolution values range from 50 for spot-style 2D codes to 200 for linear 1D bar codes to 1000 for high-resolution graphics or filled text.

Apparent resolution is also a function of the spot size of the focusing lens. The nominal spot size for an 80 mm lens is 116 microns (0.005") while a 370 mm lens produces a 540-micron (0.021") spot. This large difference in spot size must be taken into account when determining the smallest character height or 2D cell that can be marked. To obtain the best legibility when marking text, character (Text Height) values should be a minimum of seven times the focused spot size.

Delays

It is sometimes possible for the marking head's galvanometer hardware to lag slightly behind software commands while marking. To compensate, WinMark Pro provides object-specific delays. Object delays for specific objects are controlled by selecting related objects in an open drawing file and then setting appropriate delays on the *Marking* tab. See Chapter 18, *Object properties*, for further information on delay properties.

Optimization

The first step in creating a high-speed mark starts with the computer hardware, since a computer running at a faster clock speed will mark faster than one running at a slower speed. After optimizing computer hardware, mark file optimization is next. This involves determining acceptable trade-offs in WinMark Pro's Velocity, Power, and Resolution settings for a given drawing. These settings along with object delays are best determined by careful experimentation.

Vector objects

When marking vector-type objects such as stroke fonts or imported vector graphics, the fastest mark possible is sometimes not legible enough to be practical. An acceptable high-speed mark is determined by each individual application and customer. The example below suggests one approach you might take to optimize vector marks for speed:

1. Open the *Stroke demo.mkh* file created earlier.
2. Click on the text object to select it.
3. Click the *Marking* tab, increase Velocity from *30* in./sec. to *80* in./sec., increase Off Vector Velocity from *300* in./sec. to *600* in./sec. and increase Power from 60% to *75*%.

4 Position a sheet of anodized aluminum under the lens, ensure personnel are wearing safety glasses, and then make the mark.

5 Compare mark quality and cycle time to the initial stroke font mark.

If mark quality is unacceptable after increasing Velocity, try modifying object delays first because delays have less effect on marking speed. If mark quality still suffers after optimizing object delays, try decreasing both Velocity and Resolution. Remember to change only one parameter at a time while comparing results.

Open the *Vector demo.mkh* graphic file, make similar changes and compare those mark results to the original marks.

Raster objects

When marking raster-type objects such as filled TrueType fonts or imported bitmap graphics, the level of quality acceptable for a mark is again a subjective issue. The highest quality mark possible is often not fast enough from a production standpoint. As with speed, creating a quality raster mark involves trade-offs in Velocity, Power, Resolution, and delay settings and again is best determined by experimentation.

The following example suggests one approach you might take to optimize raster marks to obtain the best quality:

1 Open the *Filled TrueType demo.mkh* file created earlier.

2 Select the text object.

3 Click the *Marking* tab and decrease Velocity to *15* in./sec., Power to *60*%, and increase Resolution to *1000*.

4 Click the *Format* tab and increase Text Curve Detail to *1000* (the same as the Resolution), and set Extra Char Spacing to *0.015* inches.

5 Position a sheet of anodized aluminum under the lens, then make the mark.

6 Compare mark quality and cycle time to the initial filled TrueType mark.

If mark time is too slow, try reducing Resolution and Text Curve Detail together or try increasing Velocity and/or Power. Experiment with changing only one parameter at a time while comparing results. Open the *Bitmap demo.mkh* file and make similar changes to Velocity, Power, and Resolution. Compare those mark results to the original bitmap marks.

10 Marking parts

The *Marking parts* chapter includes subsections:

- The mark session
- Piece marking
- Batch marking

The mark session

Throughout this manual, the term "mark session" used. In WinMark Pro terms, a mark session begins after pressing F1 on the keyboard, or clicking *Start-F1* in the *Synrad WinMark* dialog box or Launcher window and ends when the Mark Count property is satisfied (the number of parts marked equals the specified Mark Count value). In automated systems when Mark Count is set to zero (0) and marking proceeds continuously as directed by "start mark" signal inputs, the mark session does not end. When using ActiveX® control, Mark Count is typically set to 1 so that the mark session begins when WinMark's MarkDrawing method is called and ends after the mark is complete so that control is returned to the calling program.

Mark sessions can be terminated at any time by pressing ESC on the keyboard, however this actually terminates or aborts the mark session; it does not end the session. This distinction is important to understand because On After Mark Session events defined on the *Automation* tab occur only when a mark session ends, not when the session is aborted. Figure 10-1 illustrates the mark session flowchart.

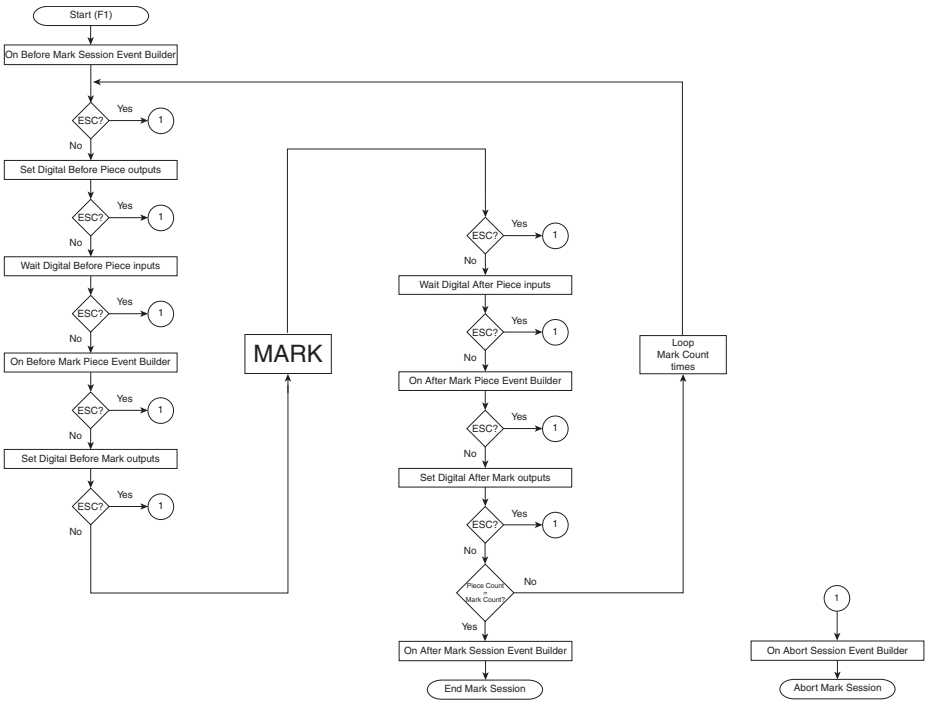


Figure 10-1 Mark session flowchart

Piece marking

In manufacturing processes that produce parts one at a time, piece marking is the most appropriate method of marking. The following procedure steps you through a sample parts marking session. This procedure may not meet your particular needs, but you can modify it by specifying different parameters.

To setup a single-piece marking operation, perform the following steps:

1 Use the *Drawing Editor* to create a file with text, artwork, or imported graphic images.

2 With the appropriate object or objects selected, set marking Velocity, Power, and Resolution on the *Marking* tab.

3 Position the cursor over a blank area of the *Drawing Canvas* and click the left mouse button or select *Drawing* in the *Object List* to deselect all objects.

4 In the *Property List*, click the *Marking* tab, and then click Mark Count. Click the arrow and set the Mark Count property to *1*.

Note: In automated piece marking applications, set Mark Count to zero (0) and carefully set input/output automation as described in Chapter 13, *Automation basics*, or our *Laser Marking FAQ*.

⚠ Danger
serious personal injury

Before performing the marking tutorials in this chapter, ensure that all personnel in the area are wearing protective eyewear. Read and follow all laser safety and fume extraction precautions described in your laser and marking head Operator's Manuals.

CO_2 lasers emit ***invisible*** infrared laser radiation at the 10.6 µm CO_2 wavelength. Since direct or diffuse laser radiation can inflict severe corneal injuries, always wear eye protection when in the same area as an exposed laser beam. Do not allow the beam to contact a person. CO_2 lasers emit an invisible beam that is capable of seriously burning human tissue.

Always be aware of the beam's path and always use a beam block while testing.

5 Place the part to be marked under the focusing lens at the point of focus.

6 On the *Toolbar*, click the *Mark* button (Figure 10-2).

Figure 10-2 Mark button on the toolbar

7 When the *Synrad WinMark* dialog box appears (Figure 10-3), click *Start-F1*.

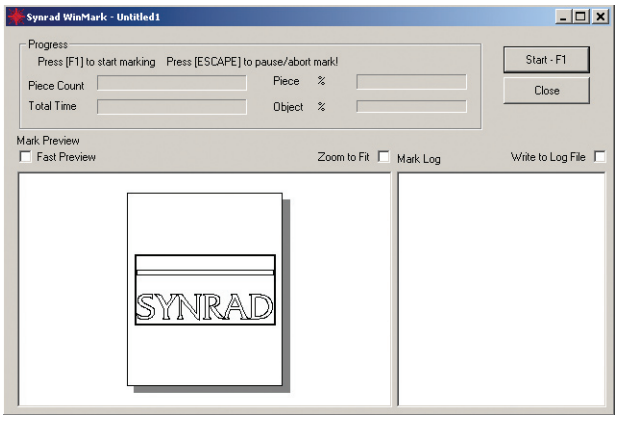


Figure 10-3 Synrad WinMark dialog box

Batch marking

If your manufacturing process produces parts in batches, trays of parts for example, batch marking is the most appropriate method for marking them. The following procedure steps you through a sample batch marking session. You will create a mark file and then use WinMark's *Array* properties to duplicate the object and create an array that matches the group of parts to be marked. This procedure may not meet your particular needs, but you can modify it by specifying different parameters.

To setup a batch marking operation, perform the following steps:

Note: Although the entire batch or tray of parts does not need to fit within the mark field, the desired mark area on all parts within the batch must fit within the marking field of the selected focusing lens.

1 Use the *Drawing Editor* to create a file with text, artwork, or imported graphic images.

2 With the appropriate object or objects selected, set marking Velocity, Power, and Resolution on the *Marking* tab.

3 In the *Property List*, click the *Format* tab, and then click Array Columns (Figure 10-4).

4 Click the arrow to increase the number of columns.

The number of columns created should match the number of columns of parts in the batch.

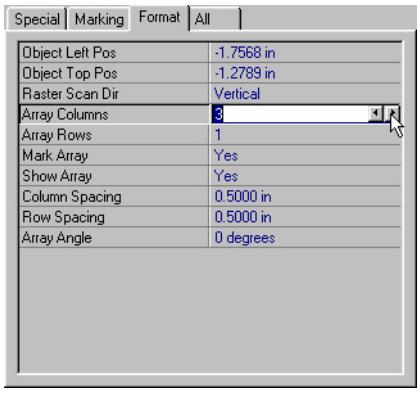


Figure 10-4 Setting number of Array Columns

5 In the *Property List*, click Array Rows (Figure 10-5), and then click the arrow to increase the number of rows.

The number of rows created should match the number of rows of parts in the batch.

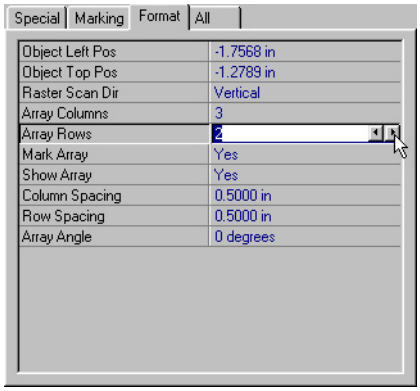


Figure 10-5 Setting number of Array Rows

6 From the *Objects* menu, click Center on Field to center the array in the marking field.

7 On the *Format* tab, adjust Column Spacing, Row Spacing, and Array Angle as required to position the arrayed image over the batch of parts.

8 Click a blank area of the *Drawing Canvas* or select *Drawing* in the *Object List* to deselect all objects.

9 In the *Property List*, click the *Marking* tab, and then click Mark Count. Click the arrow and set the Mark Count property to *1*.

Note: In automated piece marking applications, set Mark Count to zero (0) and carefully set input/output automation as described in Chapter 13, *Automation basics*, or our *Laser Marking FAQ*.

10 Place the array of parts to be marked under the focusing lens at the point of focus.

⚠ Danger
serious personal injury

Before performing the marking tutorials in this chapter, ensure that all personnel in the area are wearing protective eyewear. Read and follow all laser safety and fume extraction precautions described in your laser and marking head Operator's Manuals.

CO_2 lasers emit ***invisible*** infrared laser radiation at the 10.6 µm CO_2 wavelength. Since direct or diffuse laser radiation can inflict severe corneal injuries, always wear eye protection when in the same area as an exposed laser beam. Do not allow the beam to contact a person. CO_2 lasers emit an invisible beam that is capable of seriously burning human tissue.

Always be aware of the beam's path and always use a beam block while testing.

11 On the toolbar, click the *Mark* button (Figure 10-6).

Figure 10-6 Mark button on toolbar

12 When the *Synrad WinMark* dialog box appears (Figure 10-7), click *Start-F1*.

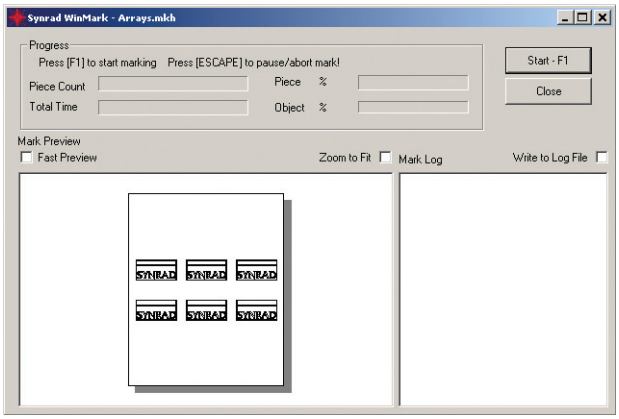


Figure 10-7 Synrad WinMark dialog box

11 Fast 2D codes

The *Fast 2D codes* chapter includes subsections:

- Introduction
- Vector-style 2D codes
- Spot-style 2D codes

Introduction

Traditionally, laser marking 2D codes is accomplished by filling square cells using a raster-scanning technique. This rasterized mark (shown in Figure 11-1) is suitable for many products requiring a code large enough to be readily identified from a distance. The primary disadvantages of raster-filled codes are longer mark times and higher heat input into the material.

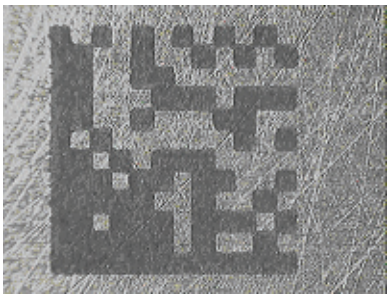


Figure 11-1 Standard raster-filled 2D code

When the job requirement calls for marking very small 2D codes, a combination of unique features in WinMark Pro allows you to create two very different (and very fast!) vector-style 2D codes. Use the "vector circle" method when creating small 2D Data Matrix or QR Codes on most materials. Use the "spot style" method when marking 2D codes on glass or printed circuit board (PCB) substrates. Overall code size is dependent upon how large or how small the focusing lens can mark individual cells.

Tip: The ratio of individual cell size to beam spot size should be on the order of 2:1 or greater.

Unlike normal (raster-filled) 2D codes, these "fast" codes have the potential for high-speed "on-the-fly" marking using FH Tracker and FH Smart marking heads at reasonably fast, continuous line speeds. In addition to high marking speeds, these vector-style 2D codes put far less heat into the material making it perfect for PCB and glass marking applications where penetration into underlying layers or uncontrolled fracturing is an issue.

Vector-style 2D codes

Figure 11-2 shows a 21-character 2D Data Matrix code marked using the "vector circle" method.

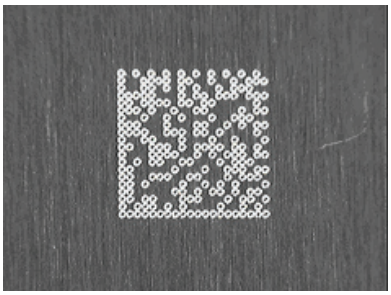


Figure 11-2 Vector-style 2D Data Matrix code

To setup a vector-style 2D code, perform the following steps:

1. Select the *2D Code Tool* and click to set an insertion point.
2. On the *Format* tab, in the 2D Barcode Text field, click the ellipsis (…). The *Text Caption Editor* dialog box appears.
3. Enter the desired alphanumeric data and click *OK*.
4. Set 2D Barcode Fixed Cell Size to *Yes*.
5. In the 2D Barcode Cell Size field, enter a value for the desired cell size.

   A good starting point is to chose a cell size that is approximately two to three times the diameter of the focused spot size.

6. Set 2D Barcode Bitmap to *No*. This changes the mark method from raster-filled squares to unfilled vector circles.
7. Set a 2D Barcode Circle Radius between 50% and 80%.

The goals of Steps 5 through 7 are to obtain an overall marked circle diameter that just fits within the cell without overlapping into adjacent cells (marking overlap can lead to Print Growth errors when scanned by barcode scanner/verification software). Figure 11-3 illustrates this concept.

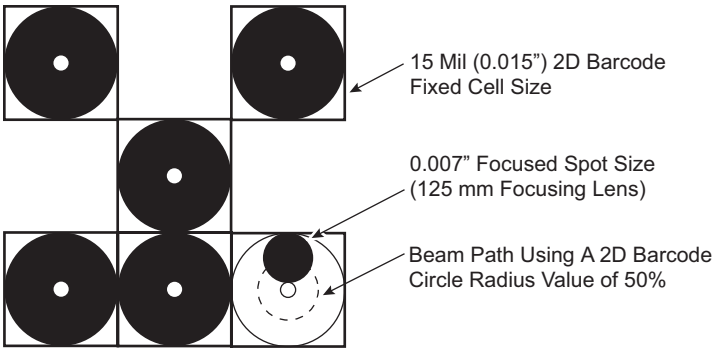

15 Mil (0.015") 2D Barcode
Fixed Cell Size
0.007" Focused Spot Size
(125 mm Focusing Lens)
Beam Path Using A 2D Barcode
Circle Radius Value of 50%

Figure 11-3 Cell size/spot size concept

8 Test mark the 2D code and, if necessary, adjust 2D Barcode Fixed Cell Size and/or 2D Barcode Circle Radius to obtain non-overlapping circular cells. If the circle outline is marked with several distinct spots instead of a smooth arc, then increase Resolution.

9 Perform a reader test of the 2D code to verify that the code is being properly scanned and that the code meets any applicable AIM (Association for Automatic Identification and Mobility) standards.

Spot-style 2D codes

Figure 11-4 shows a 2 mm × 2 mm square "spot-type" 2D Data Matrix code containing 20 alphanumeric characters.

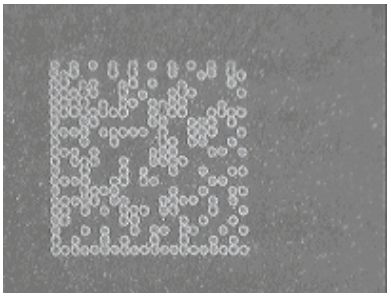


Figure 11-4 Spot-style 2D Data Matrix code

To setup a spot-style 2D code, perform the following steps:

1 Open a new drawing and set measurement units to millimeters (*Tools/General Settings…/Application Settings* – Show Units Type). Click *OK*.

2 Select the *2D Code Tool* and click to set an insertion point.

3 On the *Format* tab, in the 2D Barcode Text field, click the ellipsis (…). The *Text Caption Editor* dialog box appears.

4 Enter the desired alphanumeric data and click *OK*.

5 Set 2D Barcode Fixed Cell Size to *No*.

6 Use Table 11-1 to find the spot size for your focusing lens. Multiply lens spot size by the total number of cells across the 2D code.

Table 11-1 Lens spot size

Lens Focal Length	Spot Size ($1/e^2$), µm	Starting Resolution
FH/Fenix / DH/SH 370 mm	540	47
FH/Fenix / DH/SH 200 mm	290	88
FH/Fenix / DH/SH 125 mm	180	141
FH/Fenix 125HP*	180	141
FH/Fenix 80 mm	116	219
DH/SH 69 mm	100	254

* 125 mm lens for use with lasers 40 W and higher.

7 Divide the result from Step 6 by 1000 to convert to millimeters. This is the width and height of the 2D code.

Note: 1000 microns (µm) equals 1 mm.

8 Select the 2D code object on the *Drawing Canvas* and then from the *Objects* menu, click Transformations…. Enter the bar code size calculated in Step 7 as both *New Width* and *New Height* in the *Object Transformations* dialog box. Click *Apply*, and then click *OK*.

9 Set 2D Barcode Bitmap to *No*. This changes cell marking from raster-filled squares to unfilled vector circles.

10 Click the *Marking* tab and set the desired marking Velocity.

11 From Table 11-1, find the starting resolution for your focusing lens. Enter this number for marking Resolution.

12 Set Pline Start Delay, Pline End Delay, and Interseg Delay to *0*. Set Off Vector Delay to *500*.

13 Set Spot Marking Style to *Yes* and set Spot Mark Duration to a suitable value for the material to be marked (usually between 1 and 10).

14 Test mark the 2D code and, if necessary, adjust 2D bar code size (using the *Object Transformations* dialog box), Resolution, and Spot Mark Duration to obtain one spot per cell with equal cell spacing across the code.

15 Adjust Power to produce the desired mark. When marking glass, try a lower power setting and use multiple Mark Passes to obtain the desired mark quality.

16 Perform a reader test of the 2D code to verify that the code is properly scanned and verified.

Further improvement in mark quality or mark speed are achieved by adjusting Resolution or marking delays, primarily Pline Start Delay. To eliminate any observed tailing or ghost marking that may appear around the spot, increase Off Vector Delay slightly.

Tip: Below are sample calculations for setting up a spot-style 2D code referenced to the steps just described:

Step 4 – 2D Barcode Text is: "ABC123456789 JUNE 25, 2005". This produces a code 20 cells by 20 cells square.

Step 6 – From Table 11-1, the 80 mm FH lens spot size is 116 µm. 116 µm × 20 cells = 2320.

Step 7 – Divide 2320 by 1000. The result, 2.32 mm, is the overall width and height of the 2D code.

Step 11 – From Table 11-1, the starting Resolution value for the FH 80 mm lens is 141.

---

12 Automation basics

The *Automation basics* chapter includes subsections:

- System configuration
- Input/output configuration
- Fast Acting Safety Interlock (FASI) feature
- Developing a basic event sequence
- Automated serialization

System configuration

Additional peripheral cards

Windows PCI device management and buffering of mark data on the PCI-bus FLCC mean that there are no restrictions to adding peripheral cards for networking, motion control, etc. to the marking computer. It should be noted however, that running background applications while marking may reduce overall cycle times since WinMark Pro is competing with other open applications for processor time.

Important Note: For version 2.× users running Windows® 98/95 and ISA-bus FLCCs, SYNRAD highly recommends that WinMark Pro be installed on a stand-alone computer, separate from any networking, motion control, or parts handling systems. The installation of peripheral cards or software such as screen savers in this configuration may cause unintended marking behavior.

Additional I/O capability

WinMark Pro versions 4.4.0.4759 and above support expanded I/O capability using multiple PCI cards. Note that the additional PCI cards are strictly for I/O purposes, they cannot drive multiple marking heads.

Previous versions of WinMark Pro for Windows 98/95 using ISA-bus DAC or Fiber Link Controller Cards (FLCC) also support expanded I/O capability because the digital I/O commands in WinMark Pro's Event Builder provided the ability to specify a unique address location for each individual I/O command. This existed in part because DAC and ISA card addresses were user-selectable (as long as they did not conflict with other system devices) and I/O ports were directly addressable by the software. WinMark Pro versions 2.1.0.3468 and above that support both ISA- and PCI-bus FLCCs also benefit from this feature as long as the PCI-bus card is connected to the marking head and additional ISA-bus FLCCs are used strictly for I/O purposes. Refer to SYNRAD Technical Bulletin # 01 for further information on adding additional I/O using ISA-bus cards.

Important Note: If your application uses the Fast Acting Safety Interlock (FASI) feature, the signal to *Input Bit 3* (*IN3*) must be connected directly to the marking head and not to the FLCC.

Multiple marking heads

WinMark Pro version 4.0 does NOT currently support multiple marking heads using multiple PCI cards.

ISA-bus FLCCs could be modified to drive multiple marking heads in a master/slave configuration as long as all heads mark the same mark data at the same moment; independent marking in the master/slave mode is not possible.

Input/output configuration

I/O connections

Input/Output (I/O) connections are available on the marking head (or Fiber Link Controller card) so that you can connect control signals from/to parts handling equipment. WinMark Pro reads these input signals and sets output signals in order to fully automate the marking process. Tables 12-1 and 12-2 lists input/output bit designations in WinMark Pro and shows availability on FH Series Index/Tracker, FH Series Smart marking heads, and Fenix™ Laser Markers as well as older DH Series marking heads.

Table 12-1 Input bit table

WinMark Pro Input	DH Input*	FH/Fenix Input	FH Smart Input
Input Bit 0	0	IN0	IN0
Input Bit 1	1	IN1	IN1
Input Bit 2	2	IN2	IN2
Input Bit 3	3	IN3	IN3
Input Bit 4	4	—	IN4
Input Bit 5	5	—	IN5
Input Bit 6	—	—	IN6
Input Bit 7	—	—	IN7

* on the Fiber Link Controller Card (FLCC).

Table 12-2 Output bit table

WinMark Pro Output	DH Output*	FH/Fenix Output	FH Smart Output
Output Bit 0	—	—	—
Output Bit 1	1	—	—
Output Bit 2	—	—	—
Output Bit 3	3	—	—
Output Bit 4	4	OUT4	OUT4
Output Bit 5	5	OUT5	OUT5
Output Bit 6	6	OUT6	OUT6
Output Bit 7	—	OUT7	OUT7

* on the Fiber Link Controller Card (FLCC).

There are several important differences between DH (FLCC), FH/Fenix, and FH Smart I/O that must be considered. All four FH/Fenix inputs have individually isolated grounds and so can either sink or source current. FH Smart inputs also have isolated grounds with the exception of *IN7*, which shares a return (ground) with the return line of the isolated +15 VDC supply. With DH (FLCC) inputs however, only *Input Bit 5* is isolated, the other five inputs are all tied to a common ground point. In this case, the common FLCC inputs can only sink, not source, current.

All four FH/Fenix outputs have isolated grounds providing either sinking or sourcing current. On FH Smart heads, *OUT4*/*OUT5* and *OUT6*/*OUT7* share a common return (ground) line. In DH (FLCC) systems, *Output Bit 1*, *Output Bit 3*, *Output Bit 4*, and *Output Bit 5* share a common return (ground) point and only *Output Bit 6* is isolated.

See the "Technical Reference" section in your marking head's Operators Manual for detailed I/O circuit diagrams, pin designations, and signal specifications.

Voltage levels

All Fenix and FH Series marking heads require input voltage levels in the range of 15 V to 40 VDC for input/output signals connected to the head. When connecting I/O signals to the FLCC, input voltage levels are individually user-selectable by jumpers. The factory default, no shorting jumpers installed, enables the high (15–40 VDC) input voltage setting. Placing a shorting jumper across any jumper pad, JP3–JP8, changes the corresponding input to accept a low level 2 V to 6 VDC input signal.

Fast Acting Safety Interlock (FASI) feature

Many marking applications require a safety action to occur for each part that is marked. An example of this kind of action is a machine that lowers a shield over a part prior to marking and then raises the shield to advance the part as soon as the mark is complete. SYNRAD lasers are equipped with a *Remote Interlock* feature that disables laser firing—typically when a switch on an access door or safety enclosure is opened. However in marking applications where a safety interlock is frequently cycled, the *Remote Interlock*

creates an unacceptable delay. This delay occurs because after the interlock is closed, the laser's *Keyswitch* or remote keyswitch function must be cycled, which then invokes a built-in five-second delay prior to lasing. To address this issue, all FH (and older DH) Series marking heads and Fenix Laser Markers provide an optional Fast Acting Safety Interlock (FASI) function. When enabled, the FASI function prevents the laser from firing unless a high level signal is present on *Input Bit 3* (*IN3*). The marking head responds to this rising signal transition in less than 1 ms. To enable the FASI feature, refer to the "Technical Reference" section in your marking head's Operators Manual for details.

After your marking head is configured for FASI operation, you should modify each mark file as described below. By adding this modification in addition to the normal 'start mark' automation, WinMark Pro will not begin lasing until both the 'start mark' signal, typically *Input Bit 0* (*IN0*), AND *Input Bit 3* (*IN3*) are active.

1 Configure WinMark's Wait Digital Before Piece command (on the Drawing's *Automation* tab) to wait for a "Set" state on *Input Bit 3* (*IN3*) before marking begins. This synchronizes marking operation with the FASI safety feature. Normally, *Input Bit 0* (*IN0*) will all ready have been "Set" in order to wait for the 'start mark' signal. See Figure 12-1.

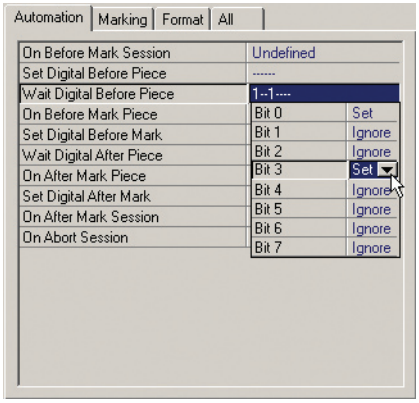


Figure 12-1 Start Mark/FASI automation setup

Note: When the FASI function is enabled, this interlock state is set in the marking head's non-volatile memory and so does not depend on WinMark Pro software or even the computer to be running. This means that even manual firing of the laser using the *Test Mark* button requires an active input signal on *IN3*.

Important Note: When using the Fast Acting Safety Interlock (FASI) feature, the signal to *Input Bit 3* (*IN3*) must be connected directly to the marking head and not to the FLCC.

Developing a basic event sequence

Automated marking and parts handling control is typically controlled by a programmable logic controller (PLC) or a computer and control program capable of sending and receiving digital signals. The marking head is controlled by WinMark Pro (or a custom Microsoft Visual C++™ or MS Visual Basic® application using WinMark Pro's ActiveX® control). WinMark Pro, through marking head or fiber link I/O, sends out-

puts to and reads inputs from a separate computer as commanded by predefined Event Builder or *Automation* tab events. Your parts handling or control system fully controls the automation sequence by setting or clearing marking head (or FLCC) input bits; WinMark responds by setting or clearing marking head (or FLCC) output bits.

The state diagram shown in Figure 12-2 below illustrates one method of defining the sequence of events necessary to coordinate parts handling and marking. Each event is numbered for cross-reference to the following steps. Refer back to Chapter 10 (Figure 10-1, *Mark Session Flowchart*) or WinMark Pro's Automation Flowchart (Help/Automation Flowchart…) to review the automation sequence.

Note: The following example assumes that an input signal from parts handling equipment is connected to *Input Bit 0* (*IN0*), and that an output signal to parts handling equipment is connected to *Output Bit 4* (*OUT4*). Although applications for WinMark Pro vary widely, the following example describes a typical automated marking operation using WaitDigital and SetDigital functions on WinMark's *Automation* tab.

WaitDigital commands force WinMark Pro to wait indefinitely for the designated ***input*** logic state to occur. Once the expected state is read on the I/O port, WinMark moves on to the next automation stage.

SetDigital commands tell WinMark Pro to write the designated ***output*** state to the I/O port. After doing so, WinMark moves on to the next automation stage.

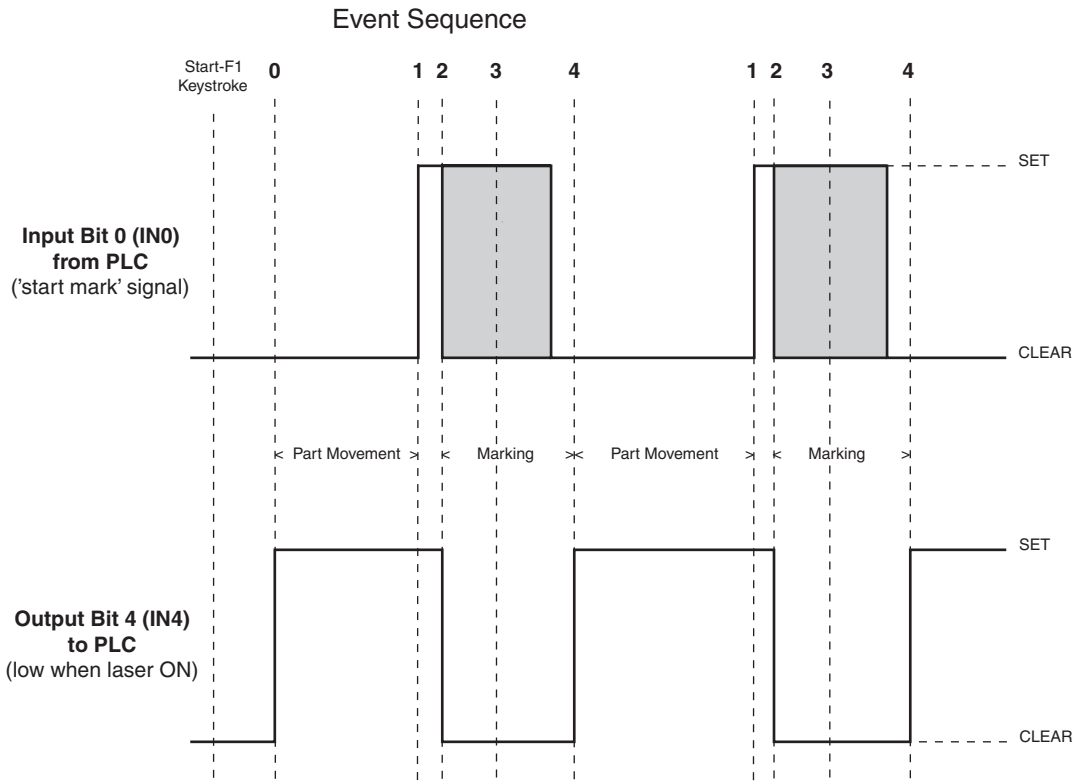


Figure 12-2 Sample state diagram

Event 0 – once the mark session is started (when the *Start-F1* button in the Launcher window is pressed), WinMark Pro begins executing the automation sequence. WinMark sets *Output Bit 4* (*OUT4*) high, signaling parts handling equipment that WinMark is ready to begin marking.

Event 1 – WinMark Pro waits indefinitely until parts handling equipment sets *Input Bit 0* (*IN0*), the 'start mark' signal, high. This action indicates that a part is in position and ready to be marked.

Event 2 – As soon as WinMark Pro recognizes the 'start mark' signal from Event 1, WinMark clears *Output Bit 4* (*OUT4*) low to inform parts handling equipment that lasing is in progress.

Event 3 – Any time after *OUT4* goes low, parts handling equipment clears *Input Bit 0* (*IN0*) low as indicated by the gray area in the state diagram. When lasing is complete, WinMark Pro verifies that the 'start mark' signal is disabled. This prevents another mark from occurring if the 'start mark' signal persists for any reason.

Event 4 – As soon as lasing is complete, WinMark Pro sets *Output Bit 4* (*OUT4*) high, signaling parts handling equipment to move another part into position.

Important Note: In order for the automation sequence to loop more than once, the Drawing's Mark Count must be greater than 1 or be set to 0 (where the mark session continues until halted by the user).

Event 0

1 Select *Drawing* from the *Object List*.

2 Select the *Automation* tab on the *Property List*.

3 Select Set Digital Before Piece and click the down arrow.

4 Click on *Bit 4*, click the down arrow, and then select *Set* from the drop-down list (Figure 12-3).

Selecting *Set* assigns a logic 1 value to *Output Bit 4* (*OUT4*). This output signals parts handling equipment that WinMark is ready for a part to be moved into position under the marking head.

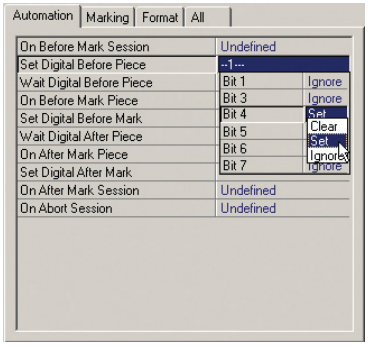


Figure 12-3 Setting "Set Digital Before Piece"

Event 1

1 Select Wait Digital Before Piece.

2 Click *Bit 0*, then select *Set* from the drop-down list (Figure 12-4).

Selecting *Set* assigns a logic 1 value as the required logic state on *Input Bit 0* (*IN0*). In this example, parts handling equipment sets *Bit 0* after a new part is in position to be marked. This input serves as the 'start mark' signal, commanding WinMark Pro to begin marking.

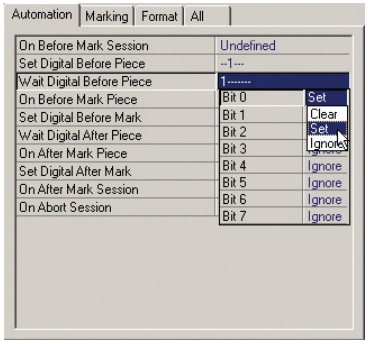


Figure 12-4 Setting "Wait Digital Before Piece"

Event 2

1 Select Set Digital Before Mark.

2 Click *Bit 4* then select *Clear* from the drop-down list (Figure 12-5).

Selecting *Clear* assigns a logic 0 value to *Output Bit 4* (*OUT4*). The digital output set here signals parts handling equipment that WinMark Pro is marking.

Important Note: WinMark Pro begins marking ***immediately*** after the Set Digital Before Mark automation step is executed.

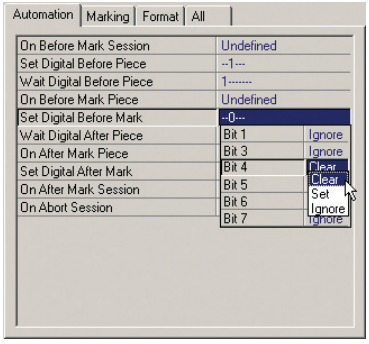


Figure 12-5 Clearing "Set Digital Before Mark"

Event 3

1 Select Wait Digital After Piece.

2 Click *Bit 0* then select *Clear* from the drop-down list (Figure 12-6).

Selecting *Clear* assigns a logic 0 value as the required state on *Input Bit 0* (*IN0*). In this example, parts handling equipment clears *Bit 0* to ensure that the 'start mark' signal is disabled before a new part is staged. To speed throughput, parts handling equipment can clear this bit any time after Event 2 occurs so that the Wait Digital After Piece test is true immediately after marking is complete.

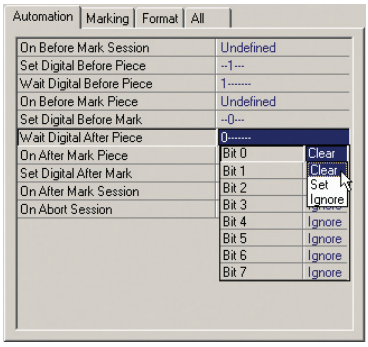


Figure 12-6 Clearing "Wait Digital After Piece"

Event 4

1 Select Set Digital After Mark.

2 Click *Bit 4* then select *Set* from the drop-down list (Figure 12-7).

Selecting *Set* assigns a logic 1 value to *Output Bit 4* (*OUT4*). The logic high value set here is an output signaling parts handling equipment that WinMark Pro has completed the mark and is ready to start the next mark when commanded.

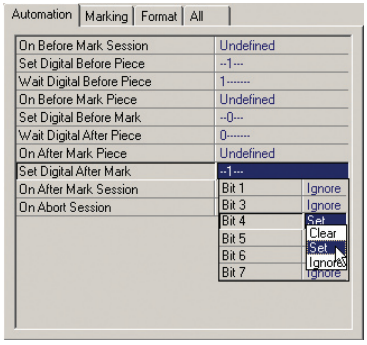


Figure 12-7 Setting "Set Digital After Mark"

Automated serialization

In addition to "communicating" or hand-shaking with parts handling equipment, WinMark Pro can also control serialization of part numbers and date or shift coding of products. The sections below describe one method for setting up automated control of part serialization.

Serial number setup

1 Select the *Text Tool* and click to set an insertion point.

2 When the *Text Caption Editor* dialog box opens, type '0000' and then click *OK*.

The number of characters entered should equal the maximum number of digits you plan to serialize. In our example, a maximum of 9999 parts would be serialized before the counter resets back to 0000.

Note: WinMark Pro does not suppress the marking of leading zeros.

Tip: In addition to serializing alphanumeric or hexadecimal text, WinMark Pro can also serialize 1D and 2D bar code data.

---

3 With the text object still highlighted, select the *Automation* tab from the *Property List* and click Auto Text Type.

4 Click the down arrow and choose Serial Number Text from the drop-down list.

5 Next, select Save Last Serial and set to *Yes*.

This ensures that the last serial number marked is saved after a mark session. The following mark session begins by retrieving the last saved number and incrementing it before marking the next piece.

6 Select Save Serial Name. The default location name for storing the last serial number is "*Slot-0*", which is located in the computer's *System Registry*. Although you cannot change the storage location, you can edit the location name to something more meaningful. By doing this, multiple mark files can have their own individual serial number strings.

7 Select Current Saved Value. If the value displayed is not zero, then zero it out (0000 in our example) to initialize the serial number counter. Figure 12-8 shows how the screen should look after this step.

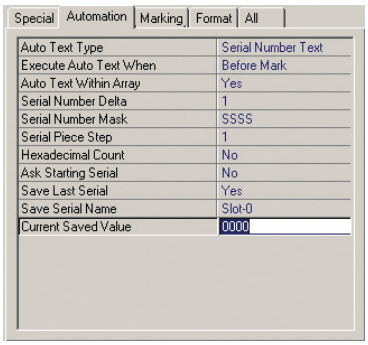


Figure 12-8 Serialization setup

Marking setup

1 Make sure the text you entered is still selected. Click the *Format* tab and go down the list, setting type characteristics as desired.

2 Next click the *Marking* tab and set desired mark characteristics including Power, Velocity, and Resolution.

3 In the *Object List*, click *Drawing* (or click outside any objects on the *Drawing Canvas*) and then click the *Making* tab.

4 Set Mark Count to zero ('0').

Figure 12-9 shows how the *Drawing's Marking* tab should look after this step. Once started, marking will continue until the ESC key is pressed. Automation signals from the control computer or PLC will determine when the laser fires. These signals will be configured in the next section.

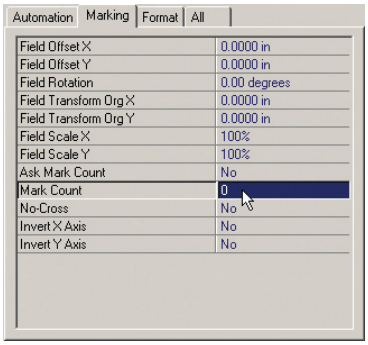


Figure 12-9 Marking setup

Automation setup

1 With the *Drawing* object still selected, click the *Automation* tab.

2 Select Set Digital Before Piece and click the down arrow.

3 Click on *Bit 4*, click the down arrow and then select *Set* from the drop-down list.

Set *Output Bit 4* to signal parts handling equipment that the laser is off and is ready for a new part.

4 Select Wait Digital Before Piece, click *Bit 0* and then select *Set* from the drop-down list.

Input Bit 0 is used as the 'start mark' signal from the PLC or automation controller. WinMark Pro's automation sequence will halt until the anticipated logic state is matched. In this example, the PLC sets *Bit 0* when all conditions for safely lasing the mark are met.

5 Select Set Digital Before Mark, click *Bit 4*, and then select *Clear*.

WinMark Pro clears this output prior to firing the laser. This bit notifies the PLC that the laser is on.

6 Select Wait Digital After Piece and clear *Bit 0*.

This input from the PLC ensures that *Bit 0* is *Clear* (the 'start mark' signal is disabled) before continuing the automation loop.

7 Select Set Digital After Mark. Click *Bit 4* and select *Set*.

Setting this output informs the PLC that marking is complete and another new part can be staged. Figure 12-10 show how the *Automation* tab should look after this step is complete.

Important Note: In WinMark Pro, when a logic state is specified for an input bit (or a combination of input bits), the next step in the *Automation List* is <u>not</u> executed until an input state matching the specified state is read on the marking head or FLCC input bus. In the Event Builder, the next event defined after a MatchDigitalState or WaitDigitalState command is <u>not</u> executed until either a matching input state is read on the input bus <u>or</u> the specified timeout period is exceeded.

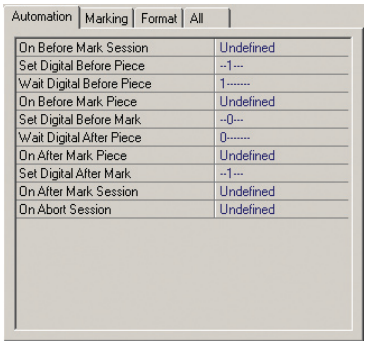


Figure 12-10 Automation setup

Note: WinMark Pro's automation loop runs in the sequence shown on the *Automation* tab. Refer back to Figure 10-1, *Mark session flowchart*, or to WinMark Pro under *Help/Automation Flowchart…* for diagrams showing how WinMark Pro steps through the automation sequence.

Begin automated marking

⚠ Danger
serious personal injury

Before performing the marking tutorials in this chapter, ensure that all personnel in the area are wearing protective eyewear. Read and follow all laser safety and fume extraction precautions described in your laser and marking head Operator's Manuals.

CO_2 lasers emit ***invisible*** infrared laser radiation at the 10.6 µm CO_2 wavelength. Since direct or diffuse laser radiation can inflict severe corneal injuries, always wear eye protection when in the same area as an exposed laser beam. Do not allow the beam to contact a person. CO_2 lasers emit an invisible beam that is capable of seriously burning human tissue.

Always be aware of the beam's path and always use a beam block while testing.

Click the *Mark* button and then click *Start-F1* when the *Synrad WinMark* dialog box (Launcher window) appears. Figure 12-11 shows the Launcher window after one iteration through the automation sequence.

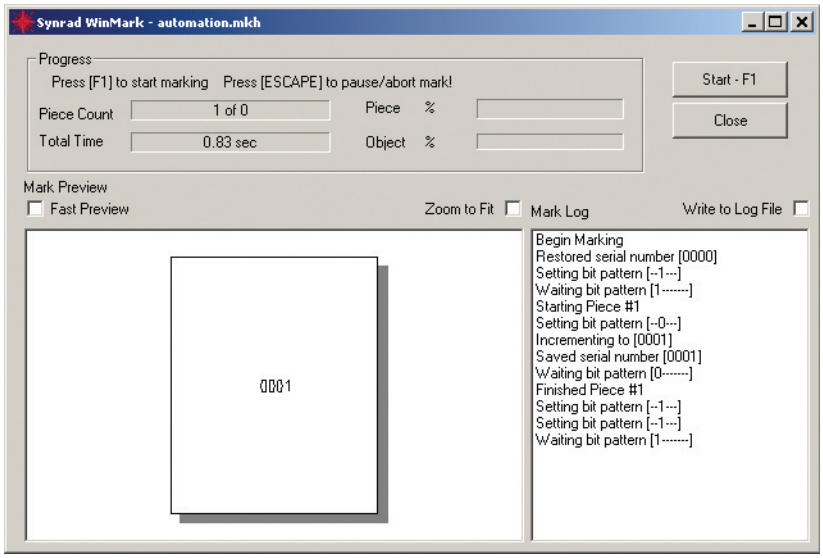


Figure 12-11 Launcher window

Note: A *Hardlock* must be installed on your computer's USB or printer port and the appropriate automation signals must be present on marking head or FLCC inputs before the laser will fire.

Important Note: If this demo is being run on a computer without a Fiber Link Controller Card (FLCC) installed, WinMark Pro will "see" all input bits as *Set* (logic high). In this case, WinMark continues executing the automation plan until a *Clear* input condition is expected.

If an FLCC is installed without input signals connected, WinMark Pro will correctly "see" all input bits as *Clear* (logic low). WinMark will continue executing the automation plan until a *Set* input condition is expected.

13 ActiveMark Automation

The *ActiveMark automation* chapter includes subsections:

- ActiveMark technology
- ActiveMark basics
- Sample VB and C++ projects

ActiveMark technology

WinMark Pro's ActiveMark™ technology transforms Windows® ActiveX®/OLE automation into a powerful user-customizable laser marking interface. After generating a mark file or template in WinMark Pro, you can create a custom ActiveX application or user-interface using Microsoft® Visual C++® or Visual Basic® development tools. This custom program uses WinMark Pro's built-in ActiveX control methods to regulate every aspect of the marking process including object placement or manipulation and communication with parts handling automation.

ActiveMark basics

Tip: When developing your automated control system using ActiveX control methods, do not place input automation into the mark file. Instead, perform any input automation functions within the ActiveX control program. By doing this, your control program remains in charge of pre-mark and post-mark automation and WinMark's MarkDrawing control method only maintains control of the operating system long enough to control laser on/off commands during the actual marking process. Performing input automation within the mark file may prevent the ActiveX control program from regaining control of system resources if the specified input state does not occur.

Tip: When developing mark files for an ActiveX environment, always set the Mark Count property to "*1*". This ensures that only a single mark is made each time the MarkDrawing method is called.

Loading the WinMark Pro ActiveMark (ActiveX) control

MS Visual Basic

The tutorial below describes how to add WinMark Pro's ActiveMark (ActiveX) control to a Visual Basic project.

To add WinMark ActiveX control to a Visual Basic project, perform the following steps:

1 Make sure that you have installed WinMark Pro. After installation, open WinMark Pro, make sure that a lens is selected, and then close WinMark Pro.

This step ensures that WinMark Pro and its associated ActiveX components are registered with the Windows operating system.

2 Open your Visual Basic project or create a new project.

3 In Visual Basic under the *Project* menu, find and select the *Components* menu item.

4 When the *Components* dialog box opens, click the *Controls* tab. If the *Selected Items Only* check box is checked, deselect it so that all registered controls are displayed.

5 Scroll through the list until you find the Synrad WinMark control item and then place a check mark to the left of the Synrad WinMark control.

If the Synrad WinMark control is not shown in the *Components* list, then the OCX has not been registered by the Windows operating system. In this case, click the *Browse* button and use the *Add ActiveX Control* dialog box to locate and *Open* the *SynMhAtx.ocx* file in the *Program Files/WinMark* folder.

6 Click *OK*. A *Toolbox* window with the Synrad WinMark ActiveX control (laser starburst icon) should appear as shown in Figure 13-1.

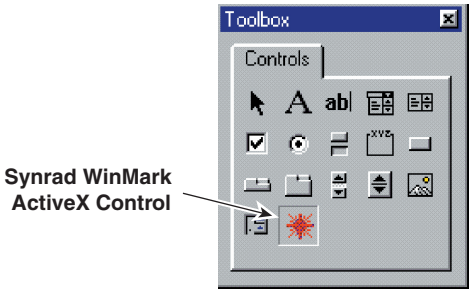


Figure 13-1 WinMark Pro's ActiveX control in VB Toolbox

7 Click on the WinMark ActiveX icon and draw the control on the form. The WinMark control appears as a preview window for marking operation.

MS Visual C++

The tutorial below describes how to add WinMark Pro's ActiveMark (ActiveX) control to the sample Visual C++ *Container* project. This same procedure is used to add the WinMark ActiveX control to any Visual C++ project or to update the ActiveX control after upgrading to a newer version of WinMark Pro.

To add WinMark ActiveX control to a Visual C++ project, perform the following steps:

1 Make sure that you have installed WinMark Pro. After installation, open WinMark Pro, make sure that a lens is selected, and then close WinMark Pro.

This step ensures that WinMark Pro and its associated ActiveX components are registered with the Windows operating system.

2 Open the *Container* program (located in the *C:\Program Files\WinMark\Samples\VisualCPlus-Sample* folder) or create a new Visual C++ project.

3 In Visual C++ under *Project/Add to Project*, find and select the *Components and Controls…* menu item.

4 From the *Components and Controls Gallery* dialog box, select *Registered ActiveX Controls*.

5 Select the *Synrad WinMark PRO Control* and then click *Insert* (see Figure 13-2).

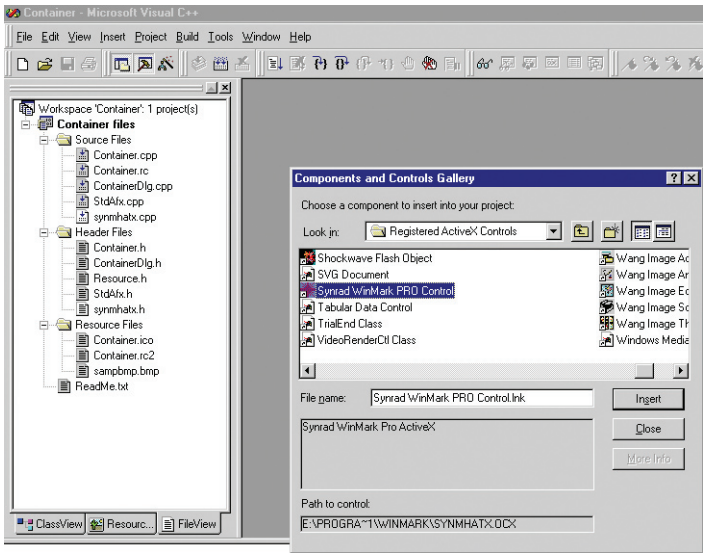


Figure 13-2 Inserting WinMark Pro's ActiveX control

6 When the *Microsoft Visual* C++ dialog box opens, click *OK* to "Insert this component".

7 When the *Confirm Classes* dialog box appears (Figure 13-3), click *OK*.

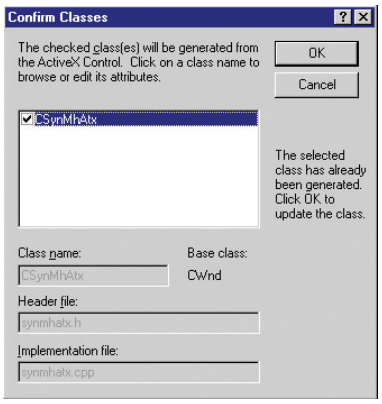


Figure 13-3 Confirm Classes dialog box

8 Click the *Close* button.

The WinMark Pro ActiveX control (*SynMhAtx.ocx*) has been added, or updated, in the open Visual C++ project. The application must be built, or rebuilt, in order to include the new control. Both the *synmhatx.cpp* and *synmhatx.h* files are machine-generated IDispatch wrapper class(es) created by MS Visual C++.

Sample VB and C++ projects

See the WinMark Pro web site at http://www.winmark.com/ActiveX_Sample_Files/activex_sample_files.htm or the WinMark Pro CD for both Visual Basic and Visual C++ sample code and mark files. These files include actual source code, instead of an executable (.exe) file therefore your development computer must contain a version of Visual Basic or Visual C++.

Important Note: We strongly recommend that all ActiveX control methods use inches as the unit of measure. Program your application or user interface to calculate the conversion to inches from other units of measure. Unpredictable object placement or operation may result if WinMark Pro's Show Units Type property is set to millimeters or centimeters.

14 Custom Test Mark feature

The *Custom Test Mark feature* chapter includes subsections:

- Test Mark Control options
- Creating a custom test mark
- Setting Test Mark Control options
- Downloading custom test mark files
- Custom test mark I/O

Customers using FH Series Index or Tracker marking heads or Fenix Laser Markers (with firmware version 7 or higher) have the ability to download a custom test mark to their marking head. Like the standard factory test mark, a custom test mark downloaded into the head's non-volatile memory is marked each time the head's *Test Mark* pushbutton is pressed. However, unlike the standard test pattern, a custom test mark allows operators to: (1) create a custom mark file with specific object properties; (2) mark the custom test mark as an Index or Tracker mark; and (3) perform true stand-alone Index or Tracker marking of the custom test mark without a computer connected to the marking head.

Access the custom test mark command in WinMark Pro by clicking *Tools* and then click Custom Test Mark… . The *Test Mark Control* dialog box (shown in Figure 14-1) opens.

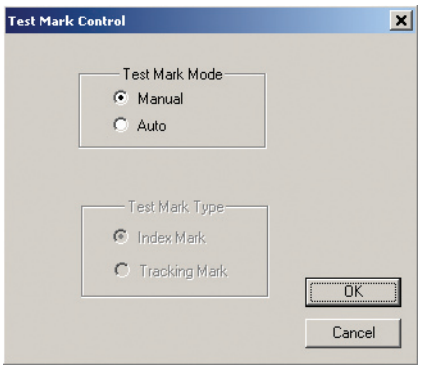


Figure 14-1 Test Mark Control dialog box

Note: WinMark Pro must be actively communicating with an FH Series marking head or Fenix Laser Marker in order to enable the custom test mark option in the *Tools* menu. If your marking head is powered up and connected to your computer, but Custom Test Mark… appears dimmed, then click the *Help* menu and choose About Synrad WinMark… . Click the *Head Info…* button and then click *OK* to exit. This operation resets communication with the head. Custom test marks are supported when the head's firmware version is 7.0 or greater, the WinMark version is 2.0.0.3019 or greater, and the Fiber Link Controller Card firmware (ISA or PCI) is v6.0 or greater.

Test Mark Control options

The custom test mark feature is controlled by selecting one of two *Test Mark Modes* and one of two *Test Mark Types*. Each control option is described below.

Test Mark Mode

Test Mark Mode determines how the custom test mark is initiated. The options, *Manual* and *Auto*, are described below.

Manual

In *Manual Test Mark Mode*, the custom test mark simply replaces the default factory test pattern in memory. In this mode, the custom test mark is always marked as a static (Index) mark, firing only when the *Test Mark* pushbutton on the marking head is pressed.

Auto

In *Auto Test Mark Mode*, the custom test mark replaces the factory test pattern in memory. Depending upon the setting of *Test Mark Type*, the stored custom test mark fires as an Index or Tracker mark when an input signal is sensed on *Input Bit 0* (*IN0*). You can also fire this test mark as an Index mark by pressing the *Test Mark* pushbutton.

Test Mark Type

Test Mark Type determines whether the custom test mark is marked as an Index or Tracker mark. *Test Mark Type* options, *Index Mark* and *Tracking Mark*, are described below. If *Manual Test Mark Mode* is selected, then *Test Mark Type* options do not apply and appear dimmed.

Index Mark

If *Test Mark Mode* is set to *Auto* and *Test Mark Type* is set to *Index* then a ***static*** custom test mark fires when the *Test Mark* pushbutton is pressed *or* when the rising edge of an input signal is detected on *Input Bit 0* (*IN0*). *Index Mark* provides the ability to load a custom test mark file, disconnect the fiber optic cable and computer, and then automatically mark an Index file each time an input signal is sensed.

Tracking Mark

If *Test Mark Mode* is set to *Auto* and *Test Mark Type* is set to *Tracking* then a ***tracking*** custom test mark fires when an input (part sensor) signal is detected on *Input Bit 0* (*IN0*) (after encoder pulses satisfy the required Sensor Distance). When the *Test Mark* pushbutton is pressed, Tracker makes a static (Index) mark. *Tracking Mark* provides the ability to load a custom test mark file, disconnect the fiber optic cable and computer, and then automatically mark a tracking file each time a part moves through the marking field.

Note: The *Tracking Mark* option appears dimmed when WinMark Pro senses it is communicating with an Index marking head.

Creating a custom test mark

Use WinMark Pro to develop a custom test mark the same way you would create any other mark file for the appropriate Index or Tracker marking style. Create a drawing with properly positioned text or graphic objects and then set object properties such as Velocity, Power, Resolution, etc.

File size for custom test marks is limited by the FH head's non-volatile memory. For example, the default test pattern fills approximately half the available memory. When a custom test mark file is too large to download, WinMark displays "*The selected mark drawing does not fit in Test Mark Memory*". If this happens, try the following options to reduce file size:

A Reduce the Off Vector Resolution of mark objects in the drawing. Non-marking resolutions of 50–100 dpi are sufficient to obtain a quality mark.

B Reduce the Resolution of mark objects. 300 dpi is adequate for most marking. For graphics with only horizontal or vertical straight lines (not diagonals), 100 dpi is sufficient.

C Refine the mark. Use stroke text instead of TrueType fonts, mark fewer words, or simplify line art.

Note: Because a custom test mark is downloaded and stored in memory as a stand-alone file, it cannot perform serialization, real-time date and time stamping, general I/O, or other automation functions as it could if run using WinMark Pro. These options are available however in the FH Series Smart marking head. Visit http://www.synrad.com for further information about FH Smart.

Setting Test Mark Control options

Creation of a custom test mark can be done offline, but to set custom test mark options the FH or Fenix head must be connected to, and communicating with, a computer running WinMark Pro.

To setup custom test mark options, perform the following steps:

1 Open or create the desired file to be downloaded as a custom test mark.

2 From the *Tools* menu, click Custom Test Mark… . The *Test Mark Control* dialog (see Figure 14-1) opens.

Note: If Index or Tracker *Auto Test Mark Mode* is currently enabled, then a dialog box appears indicating that auto test marking has been disabled. This prevents an input signal from commanding the marking head to mark while a new custom test mark file is being downloaded to memory.

3 Under *Test Mark Mode*, click *Manual* or *Auto*.

4 If *Auto Test Mark Mode* is selected, then under *Test Mark Type* choose *Index Mark* or *Tracking Mark*. The *Tracking Mark* option appears dimmed if the marking head is not a Tracker model.

The type of mark selected should match the current marking method (Index or Tracking). If the *Test Mark Type* choice is not compatible with the currently selected marking mode, a dialog box reminds you to turn the Track Marking Object property (on the Drawing's *Tracking* tab) *On* or *Off*. After setting the desired marking mode, return to *Tools* and click Custom Test Mark… to reopen the *Test Mark Control* dialog box.

Downloading custom test mark files

1 Under *Tools*, click Custom Test Mark… and when the *Test Mark Control* dialog box opens, verify that the correct *Test Mark Mode* options are selected.

Note: The default *Test Mark Mode* is *Manual*. To enable *Auto Test Mark Mode* you must select the *Auto* option each time the *Test Mark Control* dialog box opens.

2 Click *OK* to download the currently active WinMark file to FH or Fenix memory.

3 After file transfer is complete, a dialog box confirms a successful download.

4 If the data transmission fails, click *OK* to acknowledge the failure and then repeat Steps 1–3. If the custom test mark feature appears dimmed after a data transmission failure, then under *Help*, click About Synrad WinMark… and then click *Head Info… .* After closing the dialog boxes, return to the *Tools* menu. The custom test mark command should be available.

The marking head can contain only one custom test mark at a time, but a new mark file can be downloaded at any time. A copy of the factory-installed test pattern (*FactoryTestMark1.mkh*) is provided in the *Program Files\WinMark* folder so that the default test mark file can be reloaded into memory as required for lens testing or mark centering.

Custom test mark I/O

⚠ Warning
serious personal injury

FH/Fenix marking head or Fiber Link Controller Card (FLCC) outputs may be set or cleared during a custom test mark download when the mark file contains Event Builder automation.

The user must take steps to ensure that this behavior does not compromise the integrity of the control system.

It is possible for an FH Series Index or Tracker marking head or Fenix Laser Marker to set or clear outputs (OUT4–OUT7) when marking a custom test mark. However, this output capability is available only when using SetDigitalState commands in WinMark Pro's Event Builder dialog. You can define Event Builder functions during the following stages of the automation loop:

On Before Mark Session – On Before Mark Piece – On After Mark Piece – On After Mark Session

Remember that in WinMark Pro's Automation loop, On Before Mark Session and On After Mark Session events run only once before/after a marking session while On Before Mark Piece and On After Mark Piece events run before/after each individual piece. In a custom test mark application however, each individual mark is considered a mark session meaning that On Before Mark Session / On After Mark Session events are also run for each piece that is marked.

It is not possible for a custom test mark to perform non-Event Builder output functions (Set Digital Before Piece, Set Digital Before Mark, or Set Digital After Mark) or to perform any type of input automation. Custom test mark input signals are sensed in only three cases: (1) for the Fast Acting Safety Interlock (FASI) input where *Input Bit 3* (*IN3*) must be active; (2) for Index marks where *Input Bit 0* (*IN0*), the part sensor input, is expected when *Test Mark Mode* is set to *Auto*; or (3) for Tracking where *IN0* is the part sensor input and *IN1/IN2* are position inputs from the rotary encoder.

15 Tracker marking

The *Tracker marking* chapter includes subsections:

- WinMark Pro Tracking tab
- Determining line speed
- Marking text
- Marking graphics
- Mark placement
- Optimization
- Summary

Unlike static laser marking systems that require parts to be at rest during marking, SYNRAD's FH Series Tracker Marking Head and Fenix™ Tracker Laser Marker are capable of marking parts "on-the-fly" as they move through the marking field at line speeds up to 200 feet per minute. Initial material samples should be tested by SYNRAD's Applications Lab to determine the optimum power, speed, and lens size settings for your application. This testing will lead to an approximate figure for expected line speed; however, line speed is greatly affected by the type of mark you decide to create. The fastest marks are generally alphanumeric text marks such as serial numbers, date codes, lot numbers, etc., but even then changes in font type or font size can make a significant difference in actual performance.

WinMark Pro Tracking tab

When WinMark Pro senses it is communicating with an FH/Fenix Tracker, it adds another tab to the *Property List*. This additional tab, *Tracking* (shown in Figure 15-1), contains the tracking properties described below.

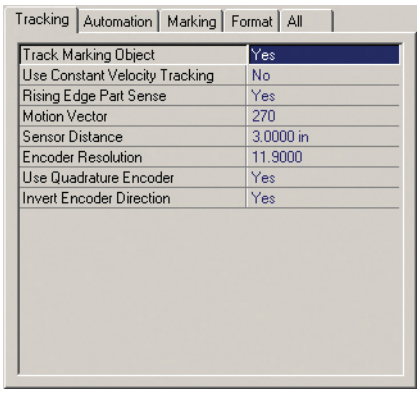


Figure 15-1 Tracking tab

Note: Newer tracking properties such as Use Constant Velocity Tracking and Use Quadrature Encoder are dependent on the firmware version installed in your marking head. Check the SYNRAD (http://www.synrad.com) or WinMark Pro (http://www.winmark.com) web sites for information on the latest released marking head firmware.

Track Marking Object

Select *Yes* to track and mark moving parts with the Tracker feature. Choose *No* to mark stationary objects.

Use Constant Velocity Tracking

Select *No* to track part movement using a rotary position encoder even as part speed varies or stops. Choose *Yes* to mark parts traveling at a constant velocity. When Constant Velocity Tracking is *Yes*, enter a fixed line speed value for the Product Line Speed property.

Rising Edge Part Sense

Set Rising Edge Part Sense based on how the part sensor output should trigger the mark. Selecting *Yes* triggers marking on the rising edge of the sensor's output waveform while selecting *No* triggers marking on the falling edge of the signal. Figure 15-2 illustrates a typical waveform from a part sensor.

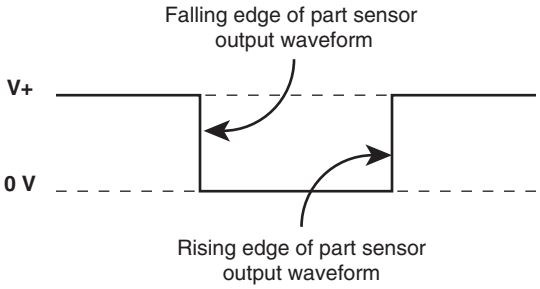


Figure 15-2 Part sensor output waveform

Motion Vector

Sets the direction of part movement through the marking field. When looking at WinMark's *Drawing Canvas* (shown in Figure 15-3), 0° is part movement towards the right, 90° is movement towards the bottom, 180° is towards the left, and 270° is part movement towards the top of the *Drawing Canvas*. The Motion Vector can be any integer value between 0 and 359 degrees.

When Track Marking Object is *Yes*, a motion arrow displays to the left of the *Drawing Canvas* showing the direction of the currently entered Motion Vector. Unmarked parts are queued up upstream of the *Marking Window* while marked pieces travel downstream (in the direction of the arrow head) after marking.

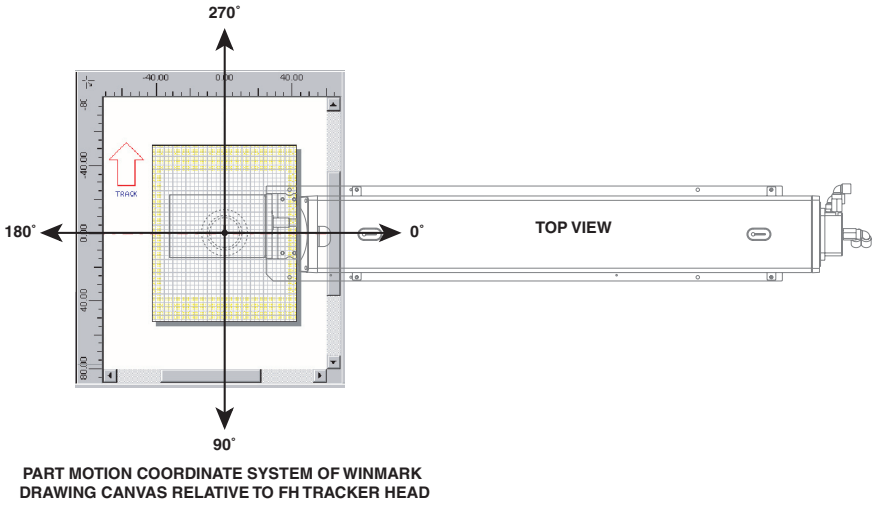


Figure 15-3 Drawing Canvas coordinates relative to Tracker

Note: Motion Vector values of 90° or 270° allow faster line speeds than 0° or 180° values because part movement through the long axis of the marking field provides a larger *Tracking Window*.

Sensor Distance

Unlike other marking methods that are time-based, Tracker uses a distance-based marking scheme. This is because at the moment a part is sensed (on the rising or falling edge of the part sense signal), the Tracker marking head begins counting rotary encoder pulses. When the number of encoder pulses counted by the head equals Sensor Distance times Encoder Resolution, marking begins. Because Tracker calculates marking vectors based on encoder pulses (distance), Tracker can continue to accurately mark an object even when it stops (once the part's *Target Area* has fully entered the *Tracking Window*).

The term, Sensor Distance, is not entirely accurate because the value entered is not related to the physical part sensor location but rather Sensor Distance is defined as ***the distance, at the moment of part sense, from the center of the mark field to the leading edge of the desired mark location***. By this definition, the "leading edge of the desired mark location" corresponds to the starting beam position, which we call the X-Y 'start mark' location. Units of measure for Sensor Distance are always entered as inches with a maximum value of 12.0 inches (305 mm). Change Sensor Distance to move the location of the mark on the part in the axis of part motion; moving the location of the mark object on the *Drawing Canvas* (in the axis of part motion) has no affect on part mark location.

Figure 15-4 illustrates the Sensor Distance concept. In this case, the part sensor is placed upstream of the mark field (parts are sensed before they reach the center of the mark field) and the sensor is set to trigger on a rising edge transition.

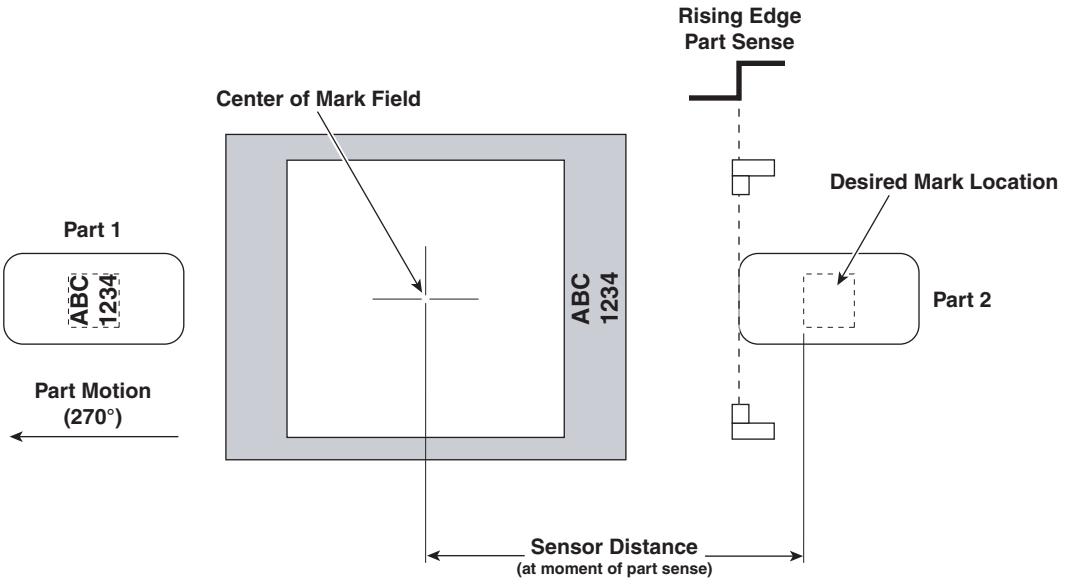


Figure 15-4 Upstream part sensor, rising edge trigger

Figure 15-5 illustrates Sensor Distance when the part sensor is positioned downstream. Even though the leading edge of the part has traveled past the center of the mark field, the desired mark location must still be upstream of the mark field's centerline.

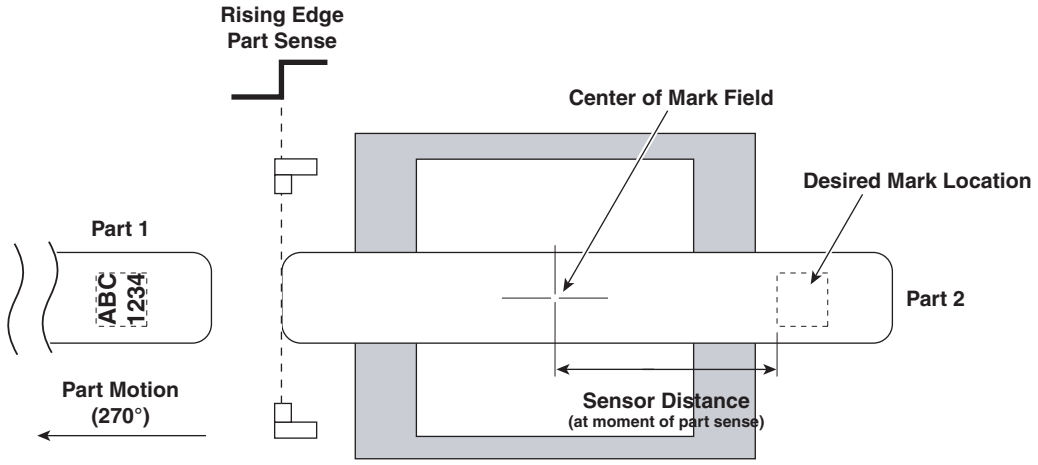


Figure 15-5 Downstream part sensor

Maximum Tracker speeds are obtained when objects are positioned near the upstream edge of WinMark's *Drawing Canvas* (to provide the largest possible *Tracking Window*) and the Motion Vector (on the *Tracking* tab) is set to 90° or 270°, which takes advantage of the FH/Fenix head's rectangular marking field.

When positioning objects on the *Drawing Canvas*, remember that each marking object has an X-Y reference point, which is the object's top left corner (in the direction of part motion). The specified Sensor Distance must be greater than or equal to the absolute value of the X-position or Y-position reference coordinate. When the Tracking Motion Vector is 90° or 270°, Sensor Distance must be greater than or equal to the absolute value of the Y-position reference point. When the Motion Vector is 0° or 180°, Sensor Distance must be greater than or equal to the absolute value of the X-position reference point. If Sensor Distance minus the X- or Y-position reference point on the *Drawing Canvas* is less than zero, this means that the part will be past its 'start mark' position before the specified Sensor Distance is traversed. To correct this "Invalid Sensor Distance" error, move the part sensor further upstream. If line speed (cycle time) is not an issue, you can instead move the object's location on the *Drawing Canvas*. In Figure 15-6, the text object's top left coordinate value is X = -0.5, Y = 2.0 (inches). If the Motion Vector is 270°, then enter a Sensor Distance of 2.0 inches or greater.

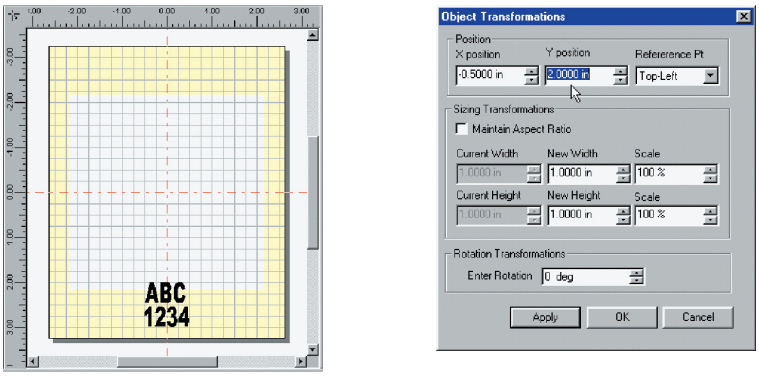


Figure 15-6 X-Y "start mark" coordinates

Rotating an object also changes the X-Y coordinates of its 'start mark' point and may require you to reposition the mark object on the *Drawing Canvas* or change Sensor Distance (move the part sensor). Depending upon the Motion Vector selected, rotating an object 180° may allow the object to be placed closer to the upstream edge of the *Drawing Canvas*, thus increasing overall line speed.

Encoder Resolution

Encoder Resolution is the number of encoder pulses received per millimeter of distance traveled by the conveyor. Encoder Resolution units are fixed as pulses/mm even when WinMark is set to display measurement units in inches or centimeters.

Use Quadrature Encoder

Select *Yes* to use a quadrature type encoder that provides both speed and direction information. Choose *No* to use an encoder that provides only speed information. Using a non-quadrature encoder may lead to mark quality issues if the product stops or reverses direction during marking.

Invert Encoder Direction

Invert Encoder Direction allows you to invert phasing of the encoder's quadrature output signal so that the actual direction of part movement through the marking field is correctly sensed as "forward". This feature eliminates the need to physically change input field wiring on the marking head.

Product Line Speed

When Constant Velocity Tracking is *Yes*, enter a constant line speed value. WinMark Pro uses this value to time the transmission of microvector data to the marking head.

Determining line speed

Definitions

Before calculating the approximate line speed that a Tracker head is capable of attaining for a given mark, several terms must be defined. To obtain the highest possible line speed and mark quality, please review these definitions carefully before designing your Tracking application.

Mark

The *Mark* is the object, or collection of objects, defined by the bounding box when all marking objects are selected in the mark file.

Marking Window

The *Marking Window* is best explained by visualizing WinMark Pro's *Drawing Canvas* superimposed on an actual part in the focal plane of the lens. The *Drawing Canvas* would exactly cover the maximum *Marking Window* for the selected lens. For example, with a 200 mm FH/Fenix lens selected, maximum *Marking Window* dimensions are 165 mm × 134 mm. When using a 125 mm FH/Fenix lens, the *Marking Window* spans a 105.6 mm × 85.7 mm area. See the *FH Series Marking Head Operator's Manual* or the *Fenix Laser Marker Operator's Manual* for complete FH/Fenix lens specifications.

Usable Field Size

Usable Field Size is the distance in millimeters from the downstream edge of *Mark* placement in the *Marking Window* to the downstream edge of the *Marking Window* in the axis of part motion. Figure 15-7 illustrates *Usable Field Size*.

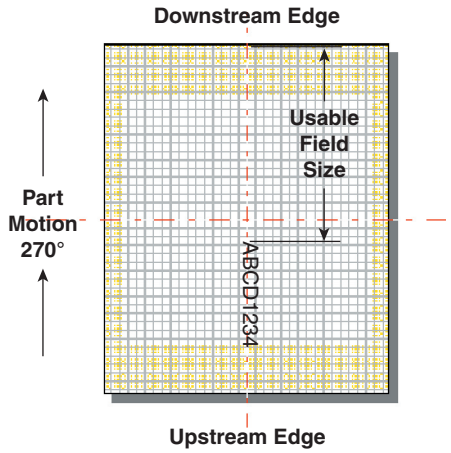


Figure 15-7 Usable Field Size

Mark Pitch

Mark Pitch is the distance, in the axis of part motion, from the leading edge of the *Mark* on one part to the leading edge of the *Mark* on the following part. ***To achieve maximum line speed, Mark Pitch must be greater than Usable Field Size***. See Figure 15-8.

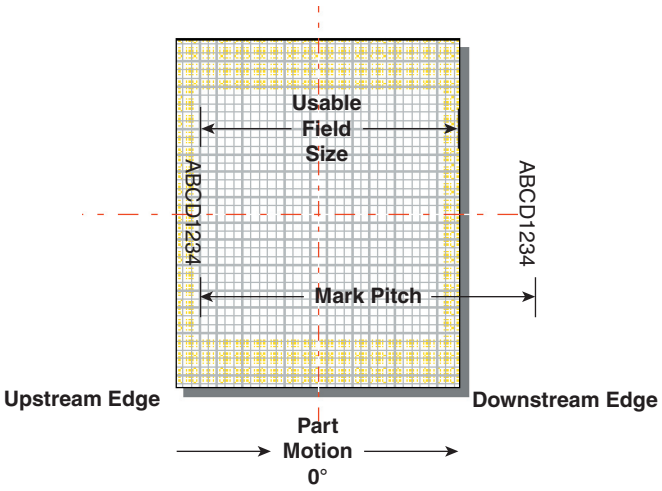


Figure 15-8 Mark Pitch greater than Usable Field Size

Cycle Time

The amount of time in seconds required to complete the *Mark* is the *Cycle Time*. Because cycle times for Index and Tracker marking are the same, marks are first tested and optimized for speed and mark quality in Index mode.

Tracking Window

The *Tracking Window* is the fixed area beneath the Tracker marking head where marking is completed without error. The length of the *Tracking Window* is defined as the smaller of either *Usable Field Size* or *Mark Pitch*.

For example, Figure 15-9 shows an application using a 200 mm lens with a 0° Motion Vector. The *Marking Window* dimension, in the direction of part motion, of 134 mm. Mark placement creates a *Usable Field Size* measuring 128 mm and part spacing creates a ***Mark Pitch*** measuring 90 mm. By definition, the *Tracking Window* is the smaller of *Usable Field Size*, which is 128 mm, or *Mark Pitch*, which measures 90 mm. In this case, the actual *Tracking Window* is limited to 90 mm.

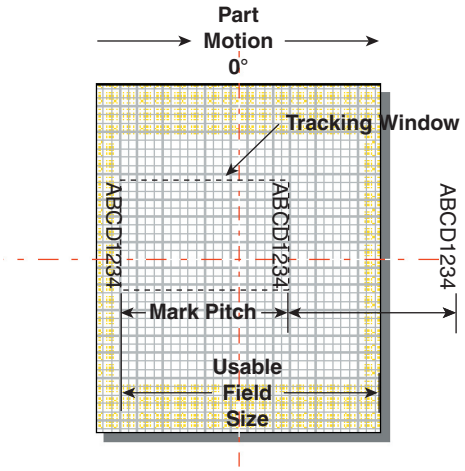


Figure 15-9 Tracking Window

Target Area

The *Target Area* is the area on the part, or array of parts, that is lased with the *Mark* as parts move under the marking head. See Figure 15-10.

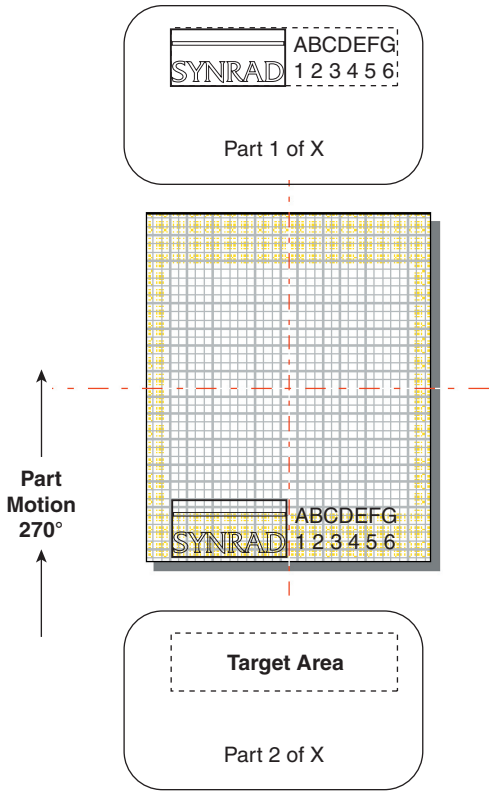


Figure 15-10 Target Area

Tracker marking criteria

- The entire *Target Area* of the part, or array of parts, must move completely within the *Tracking Window* before lasing will begin.
- Lasing must be completed on the part, or array of parts, before the next part, or array of parts, reaches the *Tracking Window*. If lasing is not complete before the *Target Area* of the next part reaches the *Tracking Window*, the next part, or array of parts, will not be marked and the mark log will display a "Line speed too fast – missed start" error.

  Depending upon Sensor Distance and *Mark Pitch*, Tracker heads can buffer up to 16 'start mark' signals. The number of start signals buffered by the head equals the number of parts that pass the part sensor before the first part enters the *Tracking Window*. Note that when a "Line speed too fast – missed start" error occurs, the buffer is reset to zero. In cases where 'start mark' signals are buffered, those pieces will not mark when the buffer resets.
- Position the part sensor so that it does not trigger while the Tracker marking head is marking a part. Refer back to Figures 15-4 and 15-5.
- Encoder Resolution, encoder pulses per millimeter of conveyor motion, must be properly set to prevent mark distortion or marking outside the *Target Area*. See the *FH Series Marking Head Operator's Manual* or the *Fenix Laser Marker Operator's Manual* for details on calculating Encoder Resolution.

After determining the correct Encoder Resolution, open and run *Linestack200.mkh* or *Linestack370.mkh* to verify the calculated value. For heads with 80 mm or 125 mm lenses, copy the linestack objects and paste them into a *New* drawing created with the correct lens setting. If Encoder Resolution is correct, then short and long lines will appear as one long line as shown in Figure 15-11.

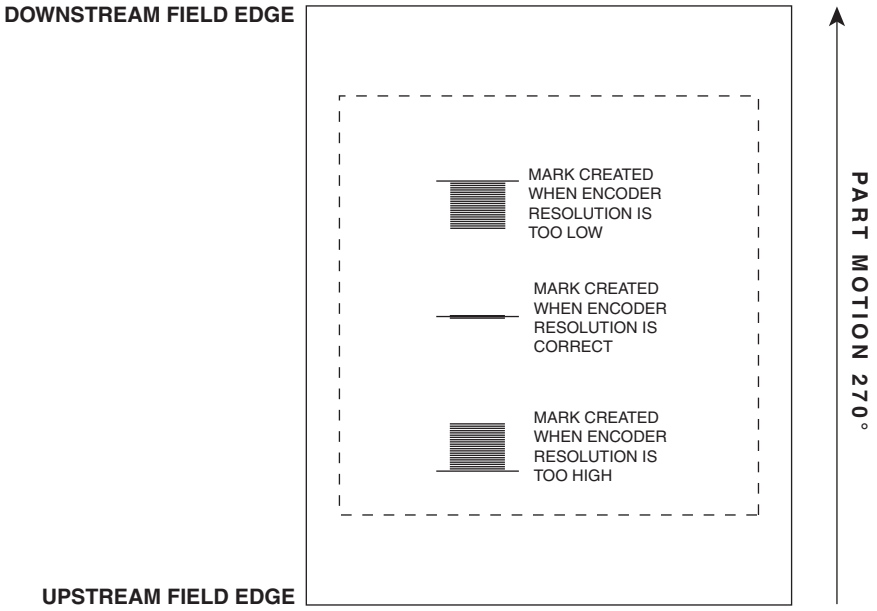

DOWNSTREAM FIELD EDGE
MARK CREATED WHEN ENCODER RESOLUTION IS TOO LOW
MARK CREATED WHEN ENCODER RESOLUTION IS CORRECT
MARK CREATED WHEN ENCODER RESOLUTION IS TOO HIGH
PART MOTION 270°
UPSTREAM FIELD EDGE

Figure 15-11 Checking Encoder Resolution

If the long line is further downstream of the short lines in the direction of part motion, then increase Encoder Resolution. If the long line is upstream of the short ones, decrease Encoder Resolution. Accuracy to the second decimal point may be required depending upon the resolution of your encoder.

- *Mark* position and orientation make a substantial difference in tracking speed because they change *Usable Field Size*. Maximum line speed is achieved when the *Mark* is positioned as close as possible to the upstream edge of the *Marking Window*. The entire mark must be positioned on the *Drawing Canvas* so that it lies within the boundaries of the *Marking Window*. Figure 15-12 shows how the same text positioned differently can give two very different values for *Usable Field Size*.

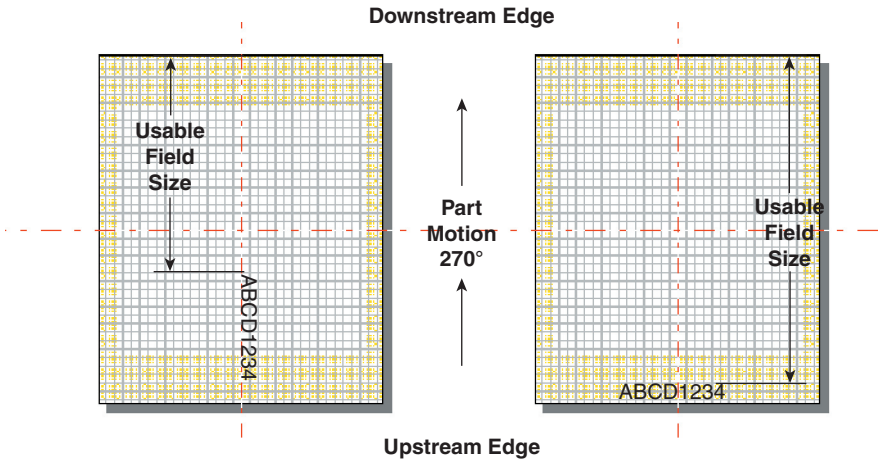


Figure 15-12 Mark length comparisons

Marked portions of the *Target Area* may move beyond the downstream edge of the *Tracking Window* as the mark is progressing; however, lasing is disabled when any unmarked microvectors move outside the *Tracking Window*. If this occurs, the mark may be missing or incomplete and the mark log will display a "Line speed too fast to finish" error. In Figure 15-13, "Line 2" will mark unless any unmarked portions of the text object move outside the *Tracking Window* before marking is complete.

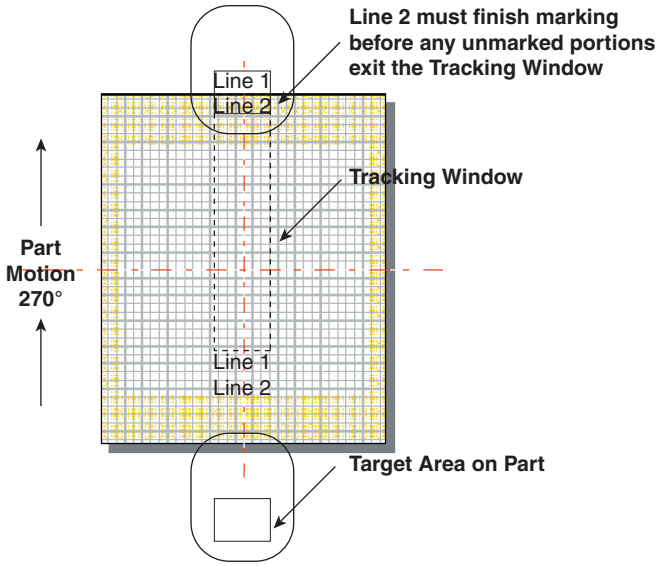


Figure 15-13 Maximum marking position in Tracking Window

Line speed formula

Note: Line speed calculations do not account for differences in computer speed or the time required for automation processes to complete their respective tasks between marks. These factors must be considered when determining the actual throughput of your production line. As with any factory automation proposal, proof-of-concept testing is highly recommended. The first step in achieving maximum line speed is to first optimize your mark file to meet mark speed and mark quality requirements for the specific material to be marked.

The formula below provides the approximate maximum line speed for a specific mark file using an FH Series Tracker or Fenix Laser Marker:

Line Speed = Tracking Window / Cycle Time

Sample calculations

Sample calculation #1 and #2 guide you through the definitions described earlier so that the correct values for Tracking parameters can be determined and inserted into the Line Speed equation.

Sample calculation #1

Refer to Figure 15-14 as you read through the following sample line speed calculation.

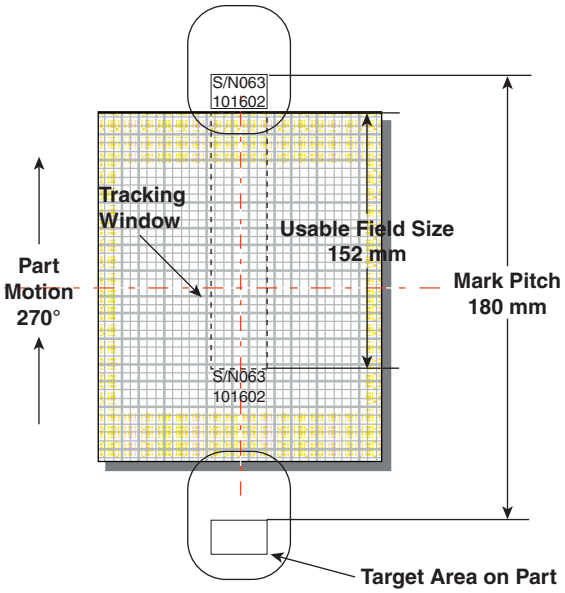


Figure 15-14 Tracker line speed calculation # 1

The mark is being made by an FH/Fenix Tracker unit using a 200 mm FH/Fenix lens; maximum lens field dimensions are 165 mm × 134 mm. A Motion Vector of 270° is set in WinMark Pro on the *Tracking* tab due to the application's part motion requirements. *Mark* placement in the *Marking Window* is such that

the *Usable Field Size* is 152 mm. The *Tracking Window* is defined as the smaller of either *Usable Field Size* or *Mark Pitch*. *Usable Field Size* is 152 mm and *Mark Pitch* is 180 mm, so the *Tracking Window* is 152 mm. *Cycle Time* for the mark (optimized in Index marking mode) is 0.32 seconds.

Line Speed = Tracking Window / Cycle Time

= 152 mm / 0.32 sec

Line Speed = 475 mm/sec = 28.5 m/min = 93.5 ft/min

Sample calculation #2

Refer to Figure 15-15 as you the following sample line speed calculation.

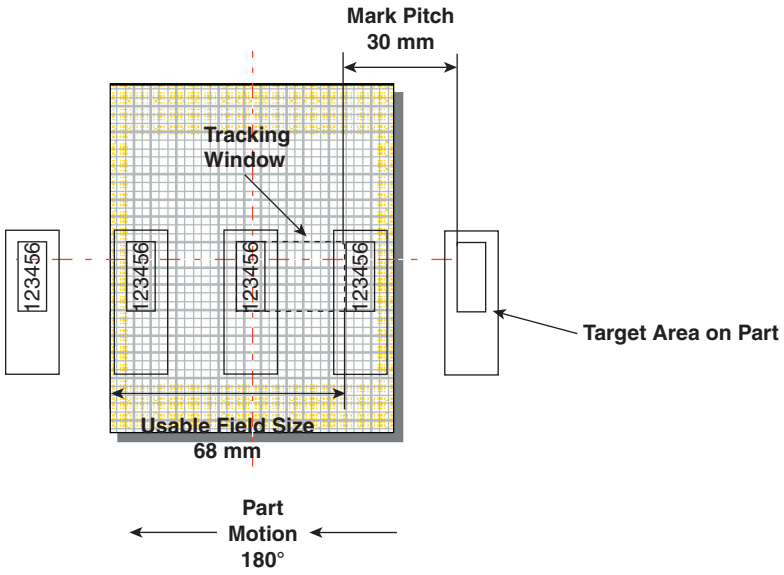


Figure 15-15 Tracker line speed calculation # 2

The mark is being made by an FH/Fenix Tracker using a 125 mm FH/Fenix lens; maximum lens field dimensions are 105.6 mm × 85.7 mm. A Motion Vector of 180° is set in WinMark on the *Tracking* tab due to the application's part motion requirements. *Usable Field Size* is 68 mm and *Mark Pitch* measures 30 mm. The *Tracking Window*, the smaller of either *Usable Field Size* or *Mark Pitch*, equals 30 mm. *Cycle Time* for the mark is 0.20 seconds.

Line Speed = Tracking Window / Cycle Time

= 30 mm / 0.20 sec

Line Speed = 150 mm/sec = 9 m/min = 29.5 ft/min

Marking text

The best, and fastest, files to mark are those files containing only vector objects like alphanumeric text created using either a stroke font or an outlined TrueType® font.

When marking a file containing just non-filled vector text (such as "123XYZ"), Tracker starts marking only when the *Target Area* encompassing "123XYZ" is fully within the *Tracking Window*. Marking can continue up until the moment the last portion of text to be marked (the last part of the "Z") exits the *Tracking Window*. Figure 15-16 illustrates placement of the text to be marked on the *Drawing Canvas*.

WinMark Pro marks text from left to right, just as you enter it in the *Text Caption Editor* dialog box; all the characters in line 1, followed by all the characters in line 2, etc. (refer back to Figure 15-13). When marking two or more lines of text, the *Mark* must not exit the *Tracking Window* before the last character on the *last* line, not the last character on the *first* line, is completed.

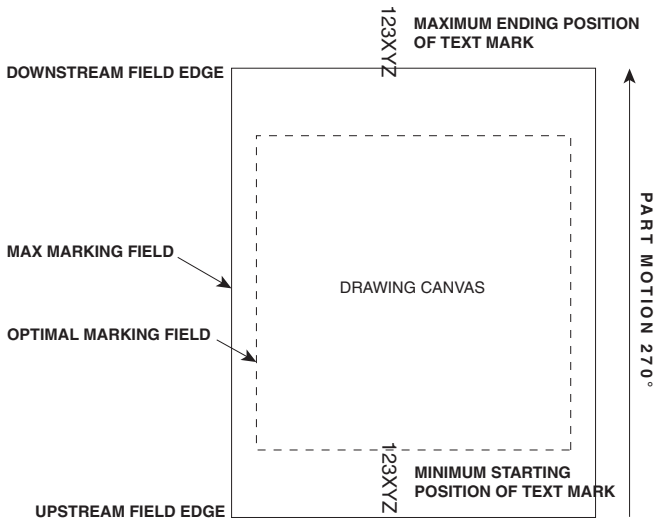


Figure 15-16 Sample text mark

Because WinMark Pro marks text from left to right and top to bottom, use this feature to ensure that marking occurs at the fastest speed possible. To achieve optimum line speeds, perform the following steps:

1 Orient either the text or the part to be marked as shown in Figure 15-17 so that the laser mark is made moving against, or counter to, the direction of part motion.

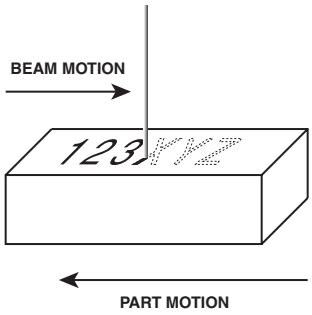


Figure 15-17 Beam/mark motion

2 Position the mark on the *Drawing Canvas* so that the entire mark is placed near the upstream edge of the *Marking Window* (Figure 15-18).

Note: Because Sensor Distance must be greater than the X or Y 'start mark' coordinate, you may need to move the physical placement of the part sensor to prevent an error when marking.

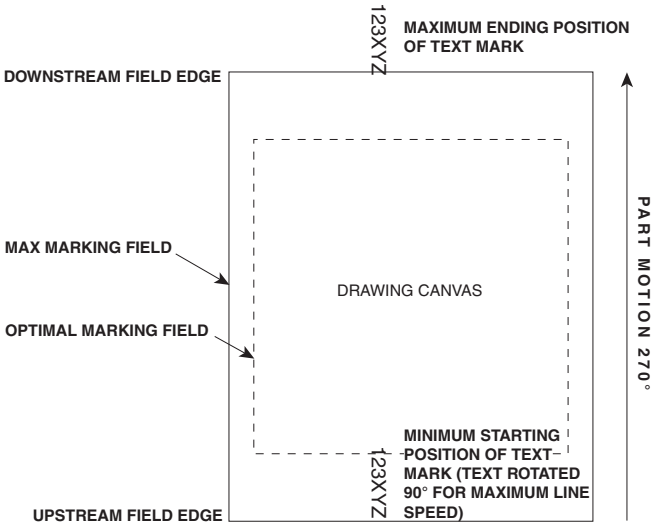


Figure 15-18 Mark orientation

3 Begin marking at the desired line speed. If line speed is set too fast, the message "*Line speed too fast to finish*" is displayed in the mark log (on the right side of the Launcher window). This error occurs when a *Target Area* containing any unmarked microvectors moves outside the *Tracking Window*.

Marking graphics

The best graphics to mark are drawings composed of unfilled vector-based artwork such as those created in drawing programs like AutoCAD®, Adobe® Illustrator®, CorelDRAW®, or Macromedia® Freehand™.

As with text objects, marking of graphic objects can continue up until the last portion of the object to be marked exits the *Tracking Window*. Be aware however, that starting points for objects such as circles and squares combined with the desired direction of part motion may not always allow the graphic to move outside the *Tracking Window* before marking is complete.

To optimize marking of files containing both text and graphics, set the mark order so that all graphic objects mark before any text objects. In WinMark, go to the *Objects* menu and click Set Marking Order. When the *Configure Marking Order* dialog box appears (Figure 15-19), arrange drawing objects as required.

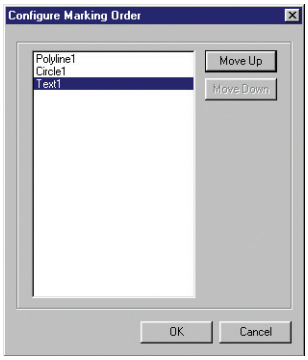


Figure 15-19 Configure Marking Order dialog box

Note: Bar codes, filled TrueType text, imported bitmap images, or other filled objects created in WinMark Pro are marked using a slower raster-scanning method. These raster objects may not be appropriate for Tracker's high-speed "on-the-fly" marking. Although 2D codes are also bitmap (raster-scanned) objects, later versions of WinMark Pro support a "fast" spot-style 2D code. This "fast" code, created by marking very small unfilled vector circles, is a viable option for marking 2D codes with Tracker marking heads.

Mark placement

To move the position of marking *along* the axis of part movement and achieve the highest potential line speed, position the *Mark* near the upstream field edge of the *Marking Window* and then adjust Sensor Distance on WinMark's *Tracking* tab to position the actual mark on the part.

To relocate the *Mark* across the axis of part motion, select mark objects on the *Drawing Canvas* and use your mouse or arrow keys to move the objects perpendicular to part motion.

Note: When positioning objects on the *Drawing Canvas*, never place any marking objects beyond the edges of the maximum *Marking Window*. Non-marking objects such as text notes for operators can be placed outside the mark field.

Optimization

Tracking variables

There are several variables related to the *Tracking Window* that can be optimized to increase marking throughput:

- Lengthen *Usable Field Size* by reducing *Mark* size.
- Rotate the *Mark* (as shown in Figure 15-12).
- If the *Tracking Window* is smaller than *Usable Field Size*, increase the *Mark Pitch* so that it is equal to, or greater than, the *Usable Field Size*. If this is not possible then consider adding a second marking head so that each head marks every other part, which effectively doubles *Mark Pitch*.

WinMark Pro property variables

Mark file cycle times are usually optimized, even in Index applications, to obtain the highest throughput speeds. To increase Tracking line speeds try the following:

- Increase Velocity and reduce Resolution.
- Increase Off Vector Velocity and reduce Off Vector Resolution.
- Use stroke text instead of TrueType fonts.
- When marking small 2D codes, set 2D Barcode Bitmap (on the *Format* tab) to *No*. This creates a code consisting of unfilled vector circles instead of filled square cells.
- Simplify line art.
- If possible, reduce the number of marking characters.
- Reduce or minimize the physical size of the Launcher window on your computer display.
- Run WinMark Pro on a faster computer.
- Load the mark file into the marking head as a Custom Test Mark. This eliminates the overhead associated with the Windows OS.

Summary

Line speed calculations do not take into account the differences in computer speeds (for instance, if the mark's *Cycle Time* is optimized on a different computer system than the one running the actual marking application), or the time required for other automation processes (such as motion controllers or automation software) to complete their respective tasks between each mark. These factors must be accounted for when determining the actual throughput of your production line. As with any factory automation proposal, proof-of-concept testing is highly recommended.

Commands & Properties

Use information in this section as a reference to WinMark Pro's powerful command and automation controls.

This section contains the following chapters:

- Chapter 16 – WinMark Pro menu commands
  explains the commands available from WinMark Pro's menu bar.
- Chapter 17 – Drawing properties
  describes drawing properties and explains how they control global drawing parameters such as tracking, input/output, and communication.
- Chapter 18 – Object properties
  describes object properties and explains how they control object display, automation, and marking.
- Chapter 19 – Event Builder commands
  describes how these predefined commands provide automation functions during the marking process.
- Chapter 20 – ActiveMark technology
  describes WinMark Pro's implementation of ActiveX/OLE control.
- Chapter 21 – Troubleshooting
  describes software-related troubleshooting issues for marking hardware and software.

Note: For specific hardware troubleshooting, please refer to your laser or your marking head Operator's Manual.

16 WinMark Pro menu commands

The *WinMark Pro menu commands* chapter includes subsections:

- File
- Edit
- Objects
- View
- Tools
- Window
- Help

WinMark Pro® displays active commands on the menu bar's drop-down lists with dark type. Commands that appear dimmed are not available until a valid object selection is made.

File

File commands (Figure 16-1) allow you to perform common file operations such as opening, saving, and printing as well as specific functions like importing graphics files or starting the laser marking process.

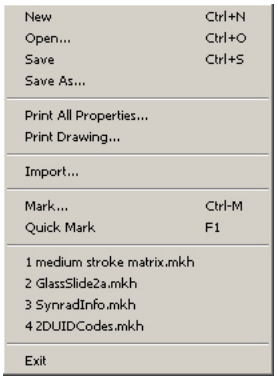


Figure 16-1 File menu

New

Open a new, untitled drawing window. You can also create a new drawing by clicking the *New* shortcut button ( ) on the toolbar.

Open...

Open the *Open* dialog box to navigate through your computer to find and open existing .mkh mark files. You can also open an existing file by clicking the *Open* shortcut button ( ) on the toolbar.

Save

Save changes to the current file. When saving a new drawing, the *Save As* dialog prompts you to enter a filename. You can also save a drawing by clicking the *Save* shortcut button ( ) on the toolbar.

Save As...

Display the *Save As* dialog, which prompts you to save the current drawing with a new filename or path.

Print All Properties...

Print object property values for all objects contained in the active drawing. Use the *Print* dialog box to specify a printer or printer options.

Print Drawing...

Print the active drawing. Use the *Print* dialog to specify a printer or printer options. You can also print by clicking the *Print Drawing* shortcut button ( ) on the toolbar.

Import...

Import a graphics file for placement in the current drawing. Click Import… to open the *Open* dialog. Find and open graphics files by filename or file extension.

WinMark Pro's Import… command relies on the File Import Filters to parse the selected file and convert the data into an importable graphic. This method is the most reliable way of opening graphics files for marking. In order to employ this approach, it is often necessary to save your graphics files in an older format (version number) so that the filters can correctly import the data. For example, when creating AutoCAD® files for WinMark Pro, *Save As* or *Export* the drawing in an AutoCAD R14 or R12 format.

In order to import vector files from more recent programs such as AutoCAD 2005 or Adobe® Illustrator® CS, WinMark Pro versions 4.6.0.4927 and above contain Copy/Paste and Drag/Drop functionality. This feature allows you to *Copy* or *Drag* a graphic from your open drawing program and *Paste* or *Drop* it onto the *Drawing Canvas*. Because graphics files often contain multiple data formats, a dialog box opens (Figure 16-2) when more than one acceptable format is available. In general, choose *Vector Image* for best marking results. In some cases, importing your graphics file through the File Import Filters may produce better results than using the Copy/Paste or Drag/Drop feature.

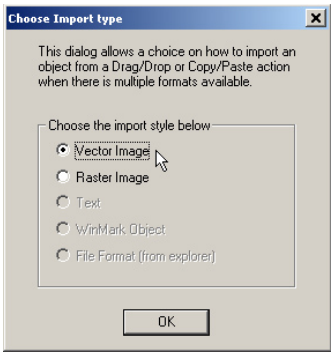

Choose Import type
This dialog allows a choice on how to import an object from a Drag/Drop or Copy/Paste action when there is multiple formats available.
Choose the import style below
Vector Image
Raster Image
Text
WinMark Object
File Format (from explorer)
OK

Figure 16-2 Choose Import Type dialog

Note: File Import Filters are automatically loaded when installing WinMark Pro versions 4.0.0.4018 and above. If you are running or have re-installed an earlier version of WinMark and wish to import vector graphic images, you must manually install the File Import Filters. File Import Filters are available from the WinMark Pro CD or the WinMark Pro web site at http://www.winmark.com.

Tip: In addition to marking monochrome (1-bit) bitmap images, WinMark Pro builds v4.0.0.4045 and above support the importation and marking of grayscale (8-bit/256 level) bitmap images. Best grayscale marks are obtained with FH Series Index and Tracker marking heads running firmware version 11 or higher. FH Smart heads can mark 1-bit monochrome images, but cannot mark 8-bit grayscale graphics. See the *Bitmap format properties* section later in this chapter for further information on marking monochrome and grayscale images.

File import notes

- Imported bitmap images are resized according to the Resolution setting specified on the *Tools/General Settings…/Object Defaults* tab. WinMark Pro does not resample bitmap images but rather resizes them to preserve the image's total pixel count. For example, the overall dimensions of a 96 DPI image imported into WinMark at 300 DPI are automatically resized to 32% of the original height and width.

- Because laser marking (with the exception of the 8-bit *Grayscale* Color Reduction Method) is a monochrome—black & white or on/off process—color or continuous-tone bitmaps are converted to 1-bit images when imported. In previous versions of WinMark Pro, the 1-bit color reduction process (*Bayer Dithering*, *Error Diffusion*, or *Halftone*) was the first step in the image transformation process; however, in WinMark Pro versions 4.0.0.4051 and above, color reduction is now applied to imported bitmap images as the last step in order to provide better raster print quality. After changing bitmap Resolution and/or resizing the image, use the Color Reduction Method property (on the *Format* tab) to change the color reduction algorithm as required. Although WinMark Pro now displays a more accurate view of actual mark quality, test marking is still recommended.

- Use the *Grayscale* Color Reduction Method to mark 8-bit grayscale images. Unlike *Bayer Dithering*, *Error Diffusion*, or *Halftone* reduction methods where color bitmaps are converted to 1-bit images, *Grayscale* reduction provides 8-bit, 256 level grayscale marking of color or continuous-tone images.

- Imported vector files are deconstructed into an entity comprised of individual objects including lines, arcs, circles, and text. If the file imports as "Group", then Ungrouping the "Group" object expands the entity into a "Polyline Set" (containing polylines and arcs) and individual "Circle" and "Text" objects. If the imported file imports as a "Polyline Set", Exploding the "Polyline Set" object expands the entity into its constituent polyline objects. In this case, elliptical or circular objects are converted to polyline objects comprised of a series of short microvectors instead of true mathematical arcs. For best marking results, create these elliptical or circular objects in WinMark Pro to obtain a true arc when marking.

- Polyline objects are optimized and grouped into a "Polyline Set" to eliminate redundant off-vector moves by the galvanometers so that cycle times are reduced. After importation, these *Polyline Sets* can be "exploded" so that individual polylines can be edited or modified as desired and then "unexploded" into a newly optimized "Polyline Set". Note that all polylines in the set are marked with the same properties, the properties of the first polyline or *Polyline Set*.

- Metafile formats (.CDR, .EPS, .EMF, .WMF, etc.) support both raster and vector objects in the same graphics file. Although vector components of a metafile are "exploded" and ungrouped into *Polyline Sets*, raster (bitmap) objects in the metafile remain whole and are not reduced to their individual components.

- In cases where a DXF file contains rotated or vertical text, the File Import Filters will rotate the text to a horizontal orientation. The best solution is to create all text using WinMark Pro instead of AutoCAD as WinMark's use of stroke and TrueType® text produces better marking results.

- When importing vector-based graphics files, WinMark Pro imports text and filled objects in addition to lines, arcs, and circles. Vector files are imported as Grouped objects that can be Ungrouped to allow editing of individual polyline or text objects. Any filled objects in the import graphic are imported as unfilled polylines and can be individually selected (after Ungrouping) and filled as required. When importing vector files containing text composed of TrueType fonts not installed on the marking computer or fonts not recognized by WinMark Pro, such as PostScript® fonts, you will need to export the file from your drawing program with text objects converted to outlines or paths in order to successfully import them into WinMark Pro.

- WinMark Pro does not currently support cross-hatching or other fill types except "solid". If you require cross-hatching to obtain the correct "look" or to increase marking speed, create individual cross-hatch lines in your drawing program. Do not fill the object using a cross-hatch (or other) pattern as WinMark will strip out these fills during importation.

Table 16-1 lists current file formats and program versions supported by WinMark Pro's File Import Filters.

Table 16-1 Supported file formats

Extension	File Format	Extension	File Format
.AI	Adobe Illustrator Format (v4.0–v7.0)	.IMG	GEM Raster Format
.BMP	Windows Bitmap	.JPG	Joint Photographic Experts Group (JPEG) File Format
.CDR	CorelDRAW! Drawing File Format (v3.0–v8.0)	.MET	OS/2 PM Metafile (v3.0)
.CGM	Computer Graphics Metafile (v1–3, ANSI, CALS, NIST)	.NAP	NAPLPS Standard ANSI Profile
.CH3	Harvard Graphics 3.0 Chart File Format (DOS)	.PCD	Kodak Photo-CD Format
.CMX	Corel ClipArt Format (v5–v7)	.PCT	Macintosh PICT Format (v2.0)
.CTM	Clear Text CGM	.PCX	Zsoft PC Paintbrush Bitmap
.DIB	MS Windows Device Independent Bitmap	.PDF	Adobe Portable Document Format (v3.0 – LZW)
.DRW	Micrografx Designer 3.1	.PDI	NAPLPS IBM PC VideoTex
.DSF	Micrografx Designer 6.0 (Win95)	.PIC	Lotus 123 Picture File Format / NAPLPS General Parametrics
.DWF	AutoDesk Drawing Web Format	.PIF	IBM Picture Interchange Format
.DWG	AutoCad Native Drawing Format (v12.0–v14.0)	.PLT	HP Graphics Language
.DXF	AutoCad Drawing Interchange Format (through v14.0)	.PNG	Portable Network Graphics Internet Format (non-LZW)
.EMF	Enhanced Windows Metafile (Win32)	.PRE	Lotus Freelance for Windows
.EPS	Encapsulated PostScript File	.PS	PostScript Language Format* (*ImageStream Level 2 Native PS Interpretation) (LZW)
.FMV	FrameMaker Vector and Raster Format	.PSD	Adobe PhotoShop Document (through v4.0)
.FPX	Kodak FlashPix File (v1.0)	.RND	AutoShade Rendering File Format
.GDF	IBM Graphics Data Format (GDDM)	.SH3	Harvard Graphics 3.0 Show
.GEM	Digital Research Graphics Environment Manager Metafile	.SY3	Harvard Graphics 3.0 Symbol File
.GIF	Graphics Interchange Format – Internet (LZW)	.TIF	Tagged Image File Format (v6)
.GP4	Group 4 CALS Format	.VSD	Visio File Format
.HGL	HP Graphics Language (Level 2.0)	.WMF	Microsoft Windows Metafile (Win16)
.HPG	HP Graphics Language (Level 2.0)	.WPG	WordPerfect Graphic File
.IGS	Initial Graphics Exchange Specification (IGES) Drawing Format (v5.1)		

Mark...

Initiate and preview marking of the current drawing through the *Synrad WinMark* dialog. The dialog box also provides piece count and total elapsed time in addition to a mark log displaying real-time marking status. You can also click the *Mark* shortcut button ( ) on the toolbar to initiate marking.

Tip: When you chose Mark… from the *File* menu or press F1, the drawing is marked immediately, but the *Launcher* window closes when the mark is complete. Use the *Mark* button on the toolbar to open *Launcher* and keep it open in cases where you want to view *Mark Log* or cycle time information or when multiple marks are required. You must press F1 or click the *Start – F1* button to begin marking after the *Launcher* window is opened using the *Mark* button.

Quick Mark

Initiate ***immediate laser marking without prompting the user***. Pressing the F1 function key also begins *Quick Mark* marking of the current drawing.

"recent files"

Display the four most recently opened files. Select a filename to open as the current drawing.

Exit

Close the WinMark Pro application. A dialog box prompts you to save any unsaved changes. Click the *Cancel* button to cancel the exit request.

Edit

Edit commands (Figure 16-3) perform common functions such as object copying, pasting, and selecting.

Figure 16-3 Edit menu

Undo

Undo the last action. The number of undo levels is controlled by WinMark's *Undo Depth* setting. Your computer memory will limit the maximum number of undo levels actually available.

Redo

Redo the previously undone action.

Cut

Cut the selection and save to the Clipboard. Only the last selection cut or copied to the Clipboard is available for use. You can also cut a selection by clicking the *Cut* shortcut ( ) on the toolbar.

Copy

Copy the selection to the Clipboard. Only the last selection cut or copied to the Clipboard is available for use. You can also copy a selection by clicking the *Copy* shortcut button ( ) on the toolbar.

Paste

Paste the current Clipboard contents into the drawing. You can also paste a selection by clicking the *Paste* shortcut button ( ) on the toolbar.

Delete

Delete the selection without saving to the Clipboard.

Duplicate

Duplicate the selection and place into the drawing.

Tip: An object inserted into a drawing using *Paste* or *Duplicate* retains all properties of the original object, however you can edit those properties individually as required.

Select All

Select all objects in the currently active drawing.

Objects

Objects commands (Figure 16-4) perform common functions such as grouping, aligning, and transforming.

Figure 16-4 Objects menu

Group

Group selected objects into a single object. When objects are grouped, WinMark Pro displays the group's default object properties although their previously defined individual mark properties are retained. Depending on the setting of Use Group Properties, all objects in a group are marked using their default group properties or are marked using their previously defined individual mark properties.

Tip: To mark a Group using individual mark properties, first set Use Group Properties to *No* (from the *Tools/General Settings…/Object Defaults* tab) and then group the desired objects.

Tip: Grouping is useful in cases where one set of objects is marked based on a specific digital or serial port input and another set of objects is marked based on another set of input conditions. For example, define a single MatchDigitalState event for a group instead of defining multiple MatchDigitalState commands for individual objects.

Ungroup

Ungroup a grouped object into its individual component objects. Ungrouped objects display all their previous user-defined properties.

Explode

When a *Polyline Set* exists, use Explode to break the set down into its constituent polylines for editing.

Unexplode

Converts selected polylines into a *Polyline Set* and optimizes polyline mark paths to obtain the fastest possible mark.

Note: All polylines in a *Polyline Set* are marked using the object properties of the first polyline or *Polyline Set* included in the selection set. Individual polyline properties are discarded and are only recoverable using the Undo command. To mark individual polylines using different mark parameters, you must Explode the *Polyline Set*.

Note: Object images may temporarily disappear from the screen display after performing an Explode or Unexplode operation. Click the *Drawing Canvas* to refresh the display window.

Align...

Align two or more selected objects either horizontally or vertically using the *Align Objects* dialog box (Figure 16-5). You can also open the *Align Objects* dialog box by clicking the *Align* shortcut ( ) on the toolbar.

Horizontal alignments are made from the selected objects' left, right, or horizontal centerlines. While Vertical alignments are made from the selected objects' top, bottom, or vertical centerlines.

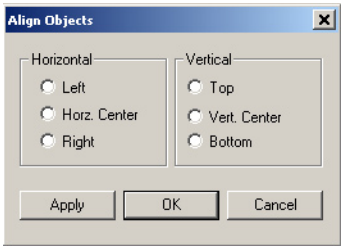


Figure 16-5 Align Objects dialog

Center on Field

Center the selection set on the 0, 0 origin point of the *Drawing Canvas*. You can also center a selection by clicking the *Center* shortcut button ( ) on the toolbar.

Transformations...

Resize, scale, rotate, or precisely move the selection set using the *Object Transformations* dialog box (Figure 16-6). X-Y position and height/width resizing values are accurate to four decimal places (0.0001"). You can also transform a selection by clicking the *Transform* shortcut ( ) on the toolbar. For ease of use, there are also *Rotate +45* ( ) and *Rotate –45* ( ) shortcut buttons located on the toolbar. Bitmap objects, including 1D and 2D barcodes, rotate only in 90-degree increments.

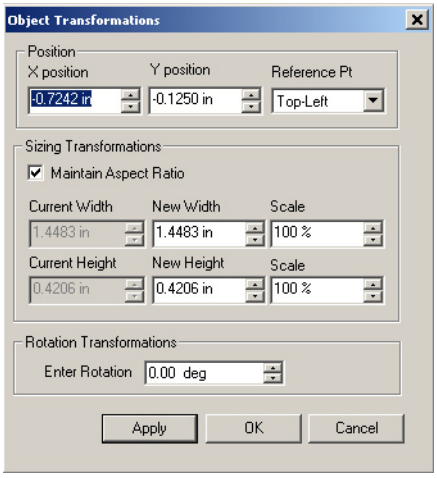


Figure 16-6 Transformations dialog

Note: For TrueType text objects, the Text Height set on the *Format* tab will not match the Object Height shown in the *Object Transformations* dialog box. The value given for the Object Transformations Object Height is the sum of the uppercase cap height plus the space provided above and below for ascenders and descenders while, for TrueType® fonts, the *Format* tab's Text Height value corresponds to the uppercase cap height of the font. Stroke font heights are variable (depending on the designer) and do not strictly conform to this description.

Tip: To obtain accurate text height, use only uppercase letters and do the following: In the *Object Transformations* dialog box, check *Maintain Aspect Ratio*, set the *Current Height* value to the desired size, and then press the TAB key to scale text. Click the *Apply* button and then click *OK*.

Mirror

Flip, or mirror, objects in the current selection set either horizontally or vertically. You can also mirror the selected objects by clicking the *Mirror X* ( ) or *Mirror Y* ( ) shortcuts on the toolbar.

Set Marking Order...

Select the order in which objects on the *Drawing Canvas* are marked. The *Configure Marking Order* dialog box displays a list of all objects in the current drawing so that you can move objects up or down in the marking order. By default, WinMark Pro marks objects in the order in which they were created or imported.

Tip: When marking sensitive materials, changing the mark order of objects helps to limit the heat input into any one area of the material at one time.

Use Object For Defaults...

Use all properties of the currently selected object as the default for any new objects created in this or future drawings. Before overwriting existing property values, a dialog box asks you to confirm this action.

View

View commands (Figure 16-7) perform common functions such as zooming in or out of the drawing or showing/hiding window elements.

Figure 16-7 View menu

Zoom Extents

Maximize the display of all objects on or off the *Drawing Canvas*. You can also "zoom all" by clicking the *Zoom Extents* shortcut button ( ) on the toolbar.

Zoom Page

Display the entire *Drawing Canvas*. You can also click the *Zoom Page* shortcut button ( ) on the toolbar to maximize your view of the canvas.

Zoom In

Zoom in towards the center of the current view by a factor of two. You can also zoom in by clicking the *Zoom In* shortcut button ( ) on the toolbar.

Zoom Out

Zoom out from the center of the current view by a factor of two. You can also zoom out by clicking the *Zoom Out* shortcut button ( ) on the toolbar.

Main Bar (Toolbar)

Toggle the check mark to show/hide the Main Bar (the toolbar).

Draw Tools

Toggle the check mark to show/hide the *Tool Box*.

Status Bar

Toggle the check mark to show/hide the *Status Bar*.

Properties

Toggle the check mark to show/hide the Properties window.

Work Book

Toggle the check mark to show/hide *Work Book* tabs. Each tab contains a separate open drawing.

Tools

Tools commands (Figure 16-8) perform functions to customize global settings and defaults or control startup options.

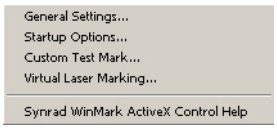


Figure 16-8 Tools menu

General Settings...

Located under General Settings… are three groups of parameters that define WinMark Pro's default Application, Drawing, and Object properties.

General Settings… Application Settings tab

Control global properties of the WinMark Pro application (see Figure 16-9).

Important Note: Properties on the Application Settings tab are not saved with the .mkh mark file. Instead, they are application properties defined for a particular WinMark Pro installation on a specific marking computer.

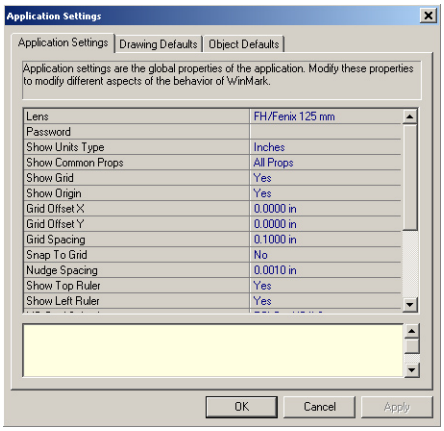


Figure 16-9 Application Settings tab

Lens

Select a lens corresponding to the currently installed focusing lens on your marking system. Click Lens and then click the ellipsis (…) to view the *Lens Selection* dialog box. When using an FH Series marking head or Fenix™ Laser Marker, select a lens from the FH/Fenix lens group. Choosing a focusing lens sets the proper Field Width and Field Height parameters for WinMark Pro and specifies the lens correction data set used by the marking head.

Password

Define a case-sensitive password to prevent unauthorized modification of .mkh mark files from WinMark Pro's *Drawing Editor*. When a password is specified, the *Enter Password* dialog box appears each time WinMark Pro is opened.

Show Units Type

Select inches, millimeters, or centimeters as the display unit for all measurements.

Note: WinMark Pro's ActiveMark™ control uses inches as the unit of measure. When running a custom application program or user-interface based on our ActiveX control methods, you must set Show Units Type to inches and then program your application to calculate the conversion from other units of measure. Unpredictable object placement or operation may result if the Show Units Type property is set to millimeters or centimeters.

Show Common Props

Display all properties or only common properties when multiple objects are selected on the *Drawing Canvas*.

Show Grid

Toggle display of *Drawing Canvas* grid lines on (*Yes*) or off (*No*).

Show Origin

Toggle display of the *Drawing Canvas X-Y Origin Lines* on (*Yes*) or off (*No*).

Grid Offset X

Offset the drawing grid in the X direction by a specified amount. This property changes only the grid display, it does not affect mark placement relative to the *Drawing Canvas*.

Grid Offset Y

Offset the drawing grid in the Y direction by a specified amount. This property changes only the grid display, it does not affect mark placement relative to the *Drawing Canvas*.

Grid Spacing

Change spacing between drawing grid lines to speed object layout while maintaining consistent spacing between elements.

Snap To Grid

Select *Yes* to constrain object placement, translation, or scaling to grid line intersections.

Nudge Spacing

Specify the distance (to 0.0001" accuracy) that selected objects are moved using the keyboard's arrow keys. If Snap to Grid is *Yes*, the distance moved is equal to the Grid Spacing value.

Show Top Ruler

Toggle display of the upper drawing ruler on (*Yes*) or off (*No*).

Show Left Ruler

Toggle display of the left drawing ruler on (*Yes*) or off (*No*).

I/O Card Selection

Select a PCI-bus Fiber Link Controller Card (FLCC) as the current marking controller. When setting up input or output automation events, choose the same card identification number (ID# 0–15) to ensure that I/O signals are properly processed. Select "*No Card Installed*" from the drop-down list to perform offline file development when an FLCC is not installed.

Note: In older versions of WinMark Pro (v1.0–v2.1) running on ISA-bus systems, the ISA-bus FLCC board is user-addressable within the range of 300–3F0 hexadecimal. In order to communicate with the marking head, convert the correct hexadecimal board address into decimal and enter this decimal value as the DA Card Address property.

Disable Marking

Toggle to disable (*Yes*) or enable (*No*) all marking output. Choose *Yes* to prevent lasing when testing or troubleshooting automated processing systems.

Undo Depth

Set the number of undo levels. Computer memory capacity will affect the maximum number of undo levels actually available.

Keyboard Locked

Lock all membrane keyboard functions (except for Fenix's *Emergency Off* pushbutton) on FH Series marking heads and Fenix Laser Markers.

Application Language

Choose one of the following languages as the default language for WinMark Pro—English, Dutch, French, German, Italian, Spanish, Japanese, or Chinese. After changing the Application Language, you must close and then re-open WinMark before the new language is displayed.

To use Japanese or Chinese languages with Windows® 98 or NT4 operating systems you must have a Windows 98/NT4 Japanese or Chinese operating system installed on your computer. To use Japanese or Chinese languages with Windows 2000 or XP operating systems you must first load the correct Chinese or Japanese codepage and set it as the default from Regional Settings in the Windows Control Panel.

Enable Driver Thread Release

Set to *No* (the default setting) for better performance in time-critical marking applications. When set to *Yes*, the WinMark Driver releases the driver thread so that other applications can run during the mark session.

Important Note: Marking performance (especially in Tracking applications) may be adversely affected if the Enable Driver Thread Release property is set to *Yes*.

Clearing Mark On Begin Mark Session

When set to *Yes*, the Clearing Mark On Begin Mark Session property initiates a clearing mark at the start of each mark session. The default setting is *No*.

Clearing Mark Interval

If Clearing Mark On Begin Mark Session is set to *No*, then select a Clearing Mark Interval. This enables a clearing mark after the specified number of marks has occurred, even across multiple mark sessions. For those customers who require a clearing mark, SYNRAD recommends an interval of once every 10,000 marks. The default is *Never*.

Important Note: The clearing mark was developed to help a small segment of customers whose mark consists of long durations of high frequency, small angle movements. This non-lasing clearing move is designed to redistribute lubricant along the galvanometer's bearing raceway and to reposition ball bearings around the race to promote a continuous and evenly worn bearing race. For more information about the how the clearing mark feature can increase the service lifetime of FH Series marking heads and Fenix Laser Markers, refer to SYNRAD Technical Bulletin # 07.

General Settings… Drawing Defaults tab

Create your own default *Drawing* properties for all new drawings you open (Figure 16-10). These properties only affect the actual marking output; *Drawing Canvas* and Launcher screen displays do not change.

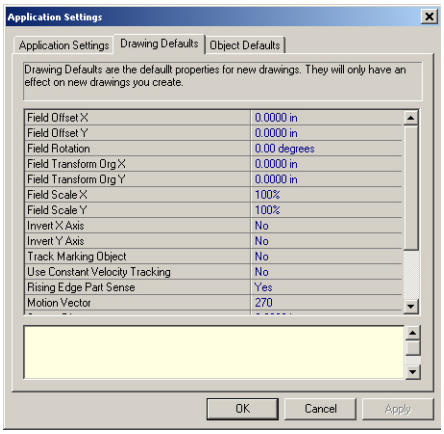

Application Settings
Application Settings
Drawing Defaults
Object Defaults
Drawing Defaults are the defaullt properties for new drawings. They will only have an effect on new drawings you create.
Field Offset X 0.0000 in
Field Offset Y 0.0000 in
Field Rotation 0.00 degrees
Field Transform Org X 0.0000 in
Field Transform Org Y 0.0000 in
Field Scale X 100%
Field Scale Y 100%
Invert X Axis No
Invert Y Axis No
Track Marking Object No
Use Constant Velocity Tracking No
Rising Edge Part Sense Yes
Motion Vector 270
OK
Cancel
Apply

Figure 16-10 Drawing Defaults tab

Field Offset X

Enter the amount of drawing offset required in the X-direction when marking.

Field Offset Y

Enter the amount of drawing offset required in the Y-direction when marking.

Field Rotation

Enter the amount of drawing rotation required around the transformation origin when marking. Field Rotation is accurate to two decimal places (0.01 degrees).

Field Transform Org X

Specify the transformation X-origin required when offsetting, rotating, or scaling a drawing for marking. The default X origin point is 0.0000 inches.

Field Transform Org Y

Specify the transformation Y-origin required when offsetting, rotating, or scaling a drawing for marking. The default Y origin point is 0.0000 inches.

Field Scale X

Specify the percentage of X-axis scaling around the transformation origin.

Field Scale Y

Specify the percentage of Y-axis scaling around the transformation origin.

Note: If Field Scale X and Field Scale Y are set to different values, the non-proportional field scaling will cause mark distortion.

Invert X Axis

Invert marking output along the X- (horizontal) axis.

Invert Y Axis

Invert marking output along the Y- (vertical) axis.

Note: When WinMark Pro senses that it is communicating with an FH Series Tracker marking head or a Fenix Tracker Laser Marker, nine tracking properties are added to the Drawing Defaults tab as shown in Figure 16-11. Refer to the *Tracker* section in your marking head's Operator's Manual for a detailed discussion of these Tracker settings.

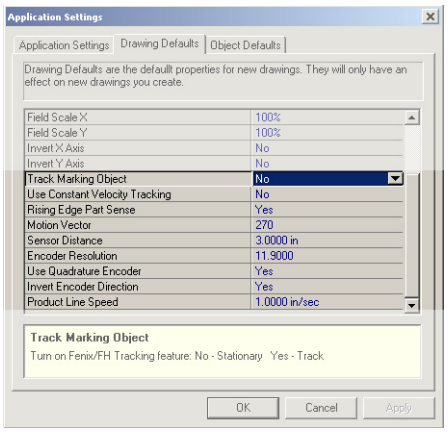

Application Settings
Application Settings
Drawing Defaults
Object Defaults
Drawing Defaults are the defaullt properties for new drawings. They will only have an effect on new drawings you create.
Field Scale X 100%
Field Scale Y 100%
Invert X Axis No
Invert Y Axis No
Track Marking Object No
Use Constant Velocity Tracking No
Rising Edge Part Sense Yes
Motion Vector 270
Sensor Distance 3.0000 in
Encoder Resolution 11.9000
Use Quadrature Encoder Yes
Invert Encoder Direction Yes
Product Line Speed 1.0000 in/sec
Track Marking Object
Turn on Fenix/FH Tracking feature: No - Stationary Yes - Track
OK
Cancel
Apply

Figure 16-11 Tracker properties on Drawing Defaults tab

In WinMark Pro version 4, Drawing Defaults (including Tracking defaults) are stored in the computer's System Registry. However, if these Tracking defaults are modified and saved with an .mkh mark file, the Tracking values in the mark file, not the Registry are used for marking each time the mark file is opened.

FH Series Smart marking heads ignore any Tracking properties stored in the mark file and instead use the values stored in the head's configuration (.cfg) file. This occurs because Tracking properties are likely to vary for each installation since it is common to see small differences in line speeds and encoder setup. The advantage of this scheme is that in cases where each installation is identical, the .cfg file from any Smart head can be uploaded, archived, and then downloaded to heads on other lines, which also aids in configuration management and/or document control.

Track Marking Object

Select *Yes* to create and mark new mark files with the Tracker feature enabled. Choose *No* to create and mark files in Index (stationary) mode.

Use Constant Velocity Tracking

When set to *No*, XY position data is generated based on input(s) from an external position encoder that varies as part motion varies. When products travel through the marking area at a constant velocity, select *Yes* to generate XY data at a fixed rate controlled by the Product Line Speed property.

Rising Edge Part Sense

Set to *Yes* to trigger on the rising edge of the part sensor's output signal. Select *No* to trigger marking on the sensor signal's falling edge.

Motion Vector

Set the direction of part movement through the marking field. When looking at WinMark Pro's *Drawing Canvas*, 0° is part movement towards the left, 90° is movement towards the bottom, 180° is towards the right, and 270° is part movement towards the top of the *Drawing Canvas*. When Track Marking Object is *Yes*, a motion arrow is displayed to the left of the *Drawing Canvas* showing the direction of the currently entered Motion Vector.

Sensor Distance

Use Sensor Distance (the distance, at the moment of part sense, from the center of the mark field to the leading edge of the desired mark location) to change the longitudinal location of the mark on the part. Sensor Distance values are entered in inches with a maximum value of 12 inches (305 mm).

Tip: Position the physical location of the part sensor so that the sensor does not trigger while the Fenix or FH head is marking a part.

Encoder Resolution

Set the number of encoder pulses counted per millimeter of distance traveled by the parts conveyor. Measurement units are fixed as pulses/mm even when WinMark is configured to display measurement units in inches or centimeters.

Use Quadrature Encoder

Choose *Yes* to specify a quadrature encoder input where 'A' and 'B' phases generate direction and velocity information. If *No*, the expected input is a velocity signal from a unidirectional encoder. This type of input is permissible in applications where the product transport does not reverse direction during marking.

Invert Encoder Direction

Toggle *Yes*/*No* to invert phasing of the position encoder's bi-directional output signal so that the actual direction of part movement through the mark field is correctly sensed as "forward". This property allows you to invert phasing of the encoder's quadrature output without physically changing field wiring.

Product Line Speed

When the Use Constant Velocity Tracking property is *Yes*, enter a Product Line Speed value (in inches per second) equal to the part velocity. Adjust Product Line Speed as required to fine-tune tracking to actual part motion.

General Settings… Object Defaults tab

Customize default object properties (Figure 16-12) for all new objects you create in a new (or current) drawing.

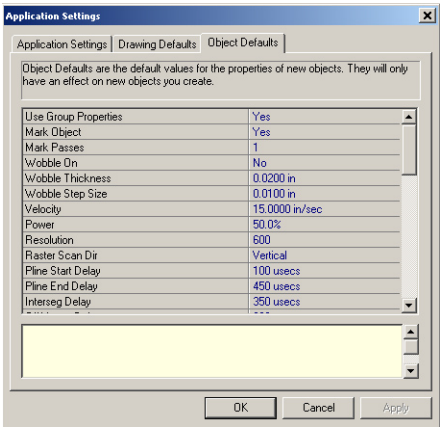


Figure 16-12 Object Defaults tab

Use Group Properties

Choose *Yes* to mark grouped entities using properties currently defined on the Object Defaults tab. Select *No* to mark grouped entities with their individual properties defined prior to grouping.

Note: In order to mark group entities using individual properties, first set Use Group Properties to *No* and then perform the Group operation.

Mark Object

Select *Yes* to mark the current selection set. Choose *No* to disable marking of selected objects.

Tip: Mark Object, a Boolean object property, is often used with MatchDigitalState automation so that a *True* (1) input condition causes the specified object or group to mark. Objects/groups are not marked if the corresponding input condition is *False* (0).

Tip: It is often useful to place marking instructions or reminders on the *Drawing Canvas* for the equipment operator. Simply enter the desired text, then size and position the text object. With only the instructional text selected, set Mark Object to *No*. Although visible, this text object will not mark and will not affect cycle times.

Mark Passes

Specify the number of passes required to mark selected objects.

Tip: Use the Mark Passes property to mark heat-sensitive materials when multiple low-power passes are required to process parts without damage.

Wobble On

Select *Yes* to mark thicker line strokes by "wobbling" the laser beam. Lines are essentially thickened by marking shorter lines at right angles to the intended line segment. Line width is determined by the value specified for Wobble Thickness and line "resolution" is determined by Wobble Step Size.

Wobble Thickness

Specify the "Wobble" width (line thickness) of selected objects.

Wobble Step Size

Enter a specific distance between "Wobble" steps. This value serves as the "resolution" of the line width—smaller steps equal higher resolution.

Velocity

Specify a "laser on" galvanometer velocity for marking the selected object.

Power

Enter the percentage of laser power required to mark the selected object.

Tip: Typically, a value of 70–80% equals the laser's specified output power rating.

Resolution

Enter a marking resolution for the selected object. Units of measure are in microvectors per inch, which is roughly analogous to the more common dots per inch (DPI) terminology.

Note: Imported bitmap images are resized according to the Resolution setting specified on the *Tools/General Settings…/Object Defaults* tab. Previously, WinMark Pro increased or decreased bitmap image size corresponding to a fixed bitmap resolution of 600 DPI. WinMark Pro does not resample bitmap images but rather resizes them to preserve the image's total pixel count. For example, the overall dimensions of a 96 DPI image imported into WinMark at 300 DPI are automatically resized to 32% of the original height and width.

Tip: In order to import a bitmap image at full scale, WinMark's Resolution setting must match the file's resolution. For example, to import a file saved at 225 DPI, first ensure that the Resolution property on the *Tools/General Settings…/Object Defaults* tab is set to *225* and then import the file.

Raster Scan Direction

Specify *Horizontal* or *Vertical* raster scanning for filled objects.

Tip: When marking raster-scanned images such as imported graphics, filled TrueType text, or barcodes, one scan direction typically marks much faster than the other depending on the object's orientation. Optimize mark cycle time by ensuring that Bi-Directional Raster is *Yes* and then test mark each filled object using first *Horizontal* and then *Vertical* Raster Scan Direction.

Pline Start Delay

Use Pline Start Delay to minimize hot spots at the beginning of a polyline or the beginning of a series of polylines. See Chapter 18, *Object properties*, for a detailed description and illustration of Pline Start Delay functionality.

Pline End Delay

Pline End Delay maintains beam output at the end of a series of polylines to ensure that the current polyline is completed before moving to the next polyline vector. See Chapter 18, *Object properties*, for a detailed description and illustration of Pline End Delay functionality.

Interseg Delay

Interseg Delay sets a delay between marking connected polylines where the end point of the current polyline is the start point of the next polyline in the object. See Chapter 18, *Object properties*, for a detailed description and illustration of Interseg Delay functionality.

Off Vector Delay

Off Vector Delay sets a proportional delay during all laser off-vector moves to eliminate "tails" when moving between non-connected polylines. See Chapter 18, *Object properties*, for a detailed description and illustration of Off Vector Delay functionality.

Bi-Directional Raster

Chose *Yes* to enable bi-directional scanning, which reduces cycle time when marking a raster image. If set to *No*, laser marking of bitmap or filled images is done in only one direction.

Note: To achieve the best image quality when Bitmap Scan Type is set to *Continuous* (or when marking Grayscale images), Bi-Directional Raster is automatically disabled; marking occurs in only one direction although you can still specify a Raster Scan Direction.

Font Type

Select either a TrueType or a stroke font for text. Use stroke fonts to mark text quickly at high throughput speeds. Use TrueType fonts to match a logo, maintain a corporate identity, or create a certain "look".

TrueType Font

Select a TrueType font for marking text. You can laser mark any TrueType font installed in the Window's *Fonts* folder.

Note: Although WinMark Pro does support TrueType (.TTF) fonts as well as the OpenType® (.TTF) fonts found on Windows 2000 and Windows XP operating systems; OpenType (.OTF) fonts are not currently supported. In addition, WinMark does not support Adobe® PostScript® (.PFM) fonts for Windows.

Stroke Font

Select one of WinMark Pro's twelve built-in stroke fonts. Fonts available include Bold, European, Gothic, LiteCom, Little, Sans Serif, Script, Simple, Trip, Tscr (Trip Script), Complex, and Simplex.

Text Height

Enter a base character height for stroke or TrueType text.

Note: For TrueType fonts, Text Height is the approximate height of uppercase letters such as "M" or "W", although this height varies slightly depending upon the metrics of the chosen typeface. In general, the Text Height for stroke fonts allows room for lowercase descenders. This means that uppercase stroke text will display and mark smaller than the value specified by Text Height.

Figure 16-13 compares both stroke and TrueType text strings. The *Drawing's* Grid Spacing is set to 0.25 inches and Text Height for both fonts is set to 1.0 inches. The *Simple* stroke font example shows that the 1.0-inch Text Height includes descenders for letters such as "g", "j", and "y". The *Times New Roman* TrueType example shows that uppercase letters are slightly less than one inch high while descenders exceed the one-inch boundary. The TrueType font's *Current Height* value in the *Object Transformations* dialog box indicates the actual overall text height is 1.21 inches.

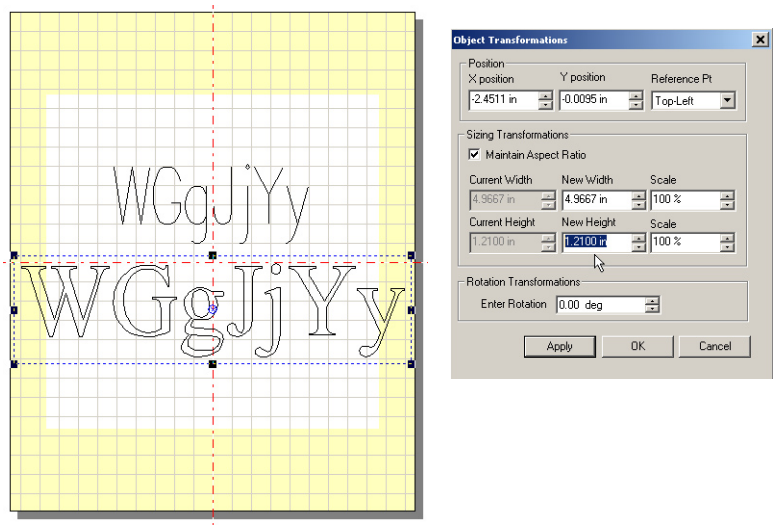


Figure 16-13 Stroke / TrueType height comparison

Tip: To obtain accurate text height, use only uppercase letters and do the following: In the *Object Transformations* dialog box, check *Maintain Aspect Ratio*, set the *Current Height* value to the desired size, and then press the TAB key to scale text. Click the *Apply* button and then click *OK*.

Text Bold

Toggle *Yes* to bold the selected TrueType text.

Text Italics

Toggle *Yes* to italicize the selected TrueType text.

Text Filled

Toggle *Yes* to fill the selected TrueType text. Choosing *Yes* causes text characters to be filled by the raster scanning method. Choosing *No* simply marks text characters as vector outlines.

Text Justification

Select *Left*, *Right*, or *Center* justification for multiline text objects.

Vector Import Scale

Select the method for scaling vector graphic files during importation. Choose from *1:1*; *X Axis*, which scales the graphic to the X (horizontal) *Drawing Canvas* extents; *Y Axis*, which scales the graphic to the Y (vertical) *Drawing Canvas* extents; or *Best Fit*, which scales the graphic to best fit the optimal mark area for the selected focusing lens.

Text Width

Enter an average width (horizontal scale) for text characters. Enter zero (*0*) to return to the selected font's default width.

Line Spacing

Enter line spacing (leading) between lines of multiline text. Enter zero (*0*) to return to the selected font's default line spacing.

Extra Char Spacing

Enter additional spacing (tracking) between text characters. Enter zero (*0*) to return to the selected font's default character spacing.

Barcode Style

Select a 1D bar code from WinMark Pro's library of over twenty common bar code types or subsets. See *Barcode format properties* in Chapter 18 for a complete list of supported bar code formats

Barcode Height

Set a specific bar code height.

Barcode Thin Width

Define a line width for thin bars in the bar code.

Tip: When setting Barcode Thin Width, do not enter a value smaller than the spot size of the selected focusing lens. Doing so causes thin bars to be marked wider than expected which encroaches on the "white" space between bars making the bar code unreadable in some cases.

Barcode Checksum

Toggle *Yes* to include a Barcode Checksum in the bar code if applicable for the Barcode Style selected. Although the checksum is appended to the bar code itself, it is not included in the human-readable text string (when Barcode Show Text is *Yes*).

Barcode Show Text

Toggle *Yes* to include human-readable text in the bar code. The text string marked is a filled (raster-scanned) TrueType font.

Tip: Human-readable bar code text always uses filled TrueType fonts. To reduce cycle times, set Barcode Show Text to *No* then create and position a new text box. Specify a stroke font for this new text object and then use the From Another Object and Assign From properties (on the *Automation* tab) to link the text object's Text Caption property to the bar code's Barcode Number value.

Barcode Text Position

If Barcode Show Text is *Yes*, select one of six locations for the placement of human-readable text around the bar code.

Barcode Text Font

Select any TrueType font installed in your computer's *Fonts* folder for the human-readable text marked as part of the bar code.

Barcode Text Size

Enter a text size for the human-readable text included with the bar code.

Barcode Thick to Thin Ratio

Select a sizing ratio between thick and thin bars/spaces in the bar code. Preset ratios include 2:1, 2.5:1, and 3:1. Specifying a larger ratio increases the width of the thick bars and the thick spaces in the code.

Bitmap Inverted

Inverts imported bitmap images so that white pixels are marked while black pixels are unmarked.

Color Reduction Method

Choose a default method for reducing black & white or color bitmaps to monochrome (1-bit) or grayscale (8-bit) images.

Bayer Dithering, *Error Diffusion*, and *Halftone* methods covert color images into monochrome (1-bit) images suitable for laser marking. The differences in the three reduction methods are in the changes made to the size, shape, and spacing of pixels in the bitmap in order to create an illusion of shading.

Use *Grayscale* to convert color bitmaps into 8-bit grayscale images. An 8-bit grayscale image provides 256 levels of gray ranging from 0 (white) to 255 (black). The best grayscale marks are obtained with FH Index or Tracker marking heads or Fenix Laser Markers running version 11, or higher, firmware.

Note: Although FH Series Smart marking heads can mark monochrome (1-bit) images, they cannot mark grayscale (8-bit) bitmaps.

Tip: Many times after choosing an initial Color Reduction Method during importation, bitmap images are resized or Resolution is altered during creation of the mark file. Because bitmap images are resolution-dependent (created for a particular display size and resolution), changing image size or resolution tends to degrade display and mark quality of the bitmap. If you must vary image size or resolution from the original settings, do so and then click Color Reduction Method. Pick the original reduction method (although in some cases, changing the reduction method produces better results) and perform a test mark. WinMark Pro applies changes in the selected Color Reduction Method to the original image in order to achieve the best possible marking results.

Bitmap Scan Type

Choose *Normal* to mark bitmaps as a sequence of microvectors with corresponding Pline delays inserted between each segment. Select *Continuous* to mark at a constant mirror speed and laser power while cycling the laser on or off in accordance with bitmap data. When the *Grayscale* Color Reduction Method is used, Bitmap Scan Type is always *Continuous*.

Note: To enhance image quality, Bi-Directional Raster is automatically disabled when Bitmap Scan Type is *Continuous* or when the Color Reduction Method is *Grayscale*.

Interseg Break Angle

Specifies a break angle between two connected polylines. No Interseg Delay is used if the angle between two marked polylines is less than the specified break angle. If the polyline angle exceeds the Interseg Break Angle, then the Interseg Delay is executed. See Chapter 18, *Object properties*, for a detailed description and illustration of Interseg Break Angle functionality.

Off Vector Velocity

Set a non-marking galvanometer velocity when traveling between polyline endpoints.

Tip: Increasing Off Vector Velocity to several times the marking Velocity will typically decrease cycle time without affecting mark quality.

Off Vector Resolution

Set a non-marking galvanometer resolution when traveling between polyline endpoints.

Spot Marking Style

When set to *Yes*, the mark path is marked using individual spots instead of a continuous line. Spot spacing is proportional to the Resolution setting. Spot Marking Style is primarily used when marking glass in order to prevent or control fracturing.

Spot Mark Duration

Set the lasing duration for the Spot Marking Style in 1/10 of a millisecond (100 microsecond) intervals.

Startup Options...

Set startup option commands (Figure 16-14) to control WinMark Launcher. Tell Launcher to open a specific drawing, to start marking immediately after opening, or to load WinMark Launcher automatically on computer startup.

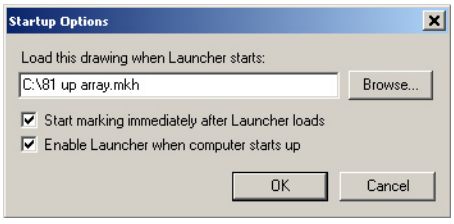


Figure 16-14 Startup Options dialog box

Custom Test Mark...

Create and load a WinMark Pro .mkh file into an FH Series Index or Tracker marking head or Fenix Laser Marker (firmware version 7 or higher), replacing the factory test mark pattern. Unlike the standard test mark, the custom test mark feature allows you to: (1) create a custom mark file with specific object properties; (2) mark the custom test mark as an Index or Tracker mark; or (3) perform standalone Index or Tracker marking initiated by an automation input, even without a computer attached to the marking head!

Custom test mark options are set in the *Test Mark Control* dialog box (Figure 16-15). The following section describes how to configure custom test marking.

Note: The Custom Test Mark... selection is only active when WinMark Pro senses it is actively communicating with an FH Series Index or Tracker marking head or a Fenix Laser Marker, otherwise the menu item is disabled (dimmed). If the FH head is powered up and connected to your marking computer and Custom Test Mark... appears dimmed under the *Tools* menu, then under *Help*, click About Synrad WinMark... and then click *Head Info* to reset communications. The custom test mark feature is supported when these three conditions are met: (1) the marking head's firmware version is v7.0 or greater, (2) WinMark Pro version 2.0.0.3019 or greater is used, and (3) Fiber Link Controller Card (FLCC) firmware is v6.0 or greater.

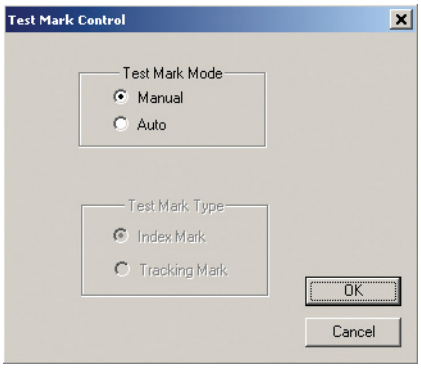


Figure 16-15 Test Mark Control dialog box

Test Mark Mode

Test Mark Mode determines how the custom test mark is initiated, either manually or automatically.

In *Manual* Test Mark Mode, the custom test mark simply replaces the default factory test pattern in non-volatile memory. In this mode, the custom test mark is always marked as a static (Index) mark, firing only when the *Test Mark* pushbutton on the FH head is pressed.

In *Auto* Test Mark Mode, the custom test mark replaces the factory test pattern in non-volatile memory. Depending on the setting of Test Mark Type, the stored custom test mark automatically fires as an Index or Tracker mark each time an input signal is sensed on *Input Bit 0* (*IN0*). The *Auto* test mark file always marks in a static (Index) mode when the *Test Mark* pushbutton on the FH head is pressed.

Test Mark Type

Test Mark Type determines whether the custom test mark is marked as an Index or Tracker mark. If *Manual* Test Mark Mode is selected, then Test Mark Type options do not apply and are dimmed.

When Test Mark Mode is set to *Auto* and Test Mark Type is set to *Index Mark*, then a static custom test mark fires when the *Test Mark* pushbutton on the marking head is pressed *or* when the rising edge of an input signal is detected on *Input Bit 0* (*IN0*). *Index Mark* provides the ability to load a custom test mark file, disconnect the fiber link cable and computer, and then automatically mark an Index file each time *Input Bit 0* (*IN0*) goes active.

When Test Mark Mode is set to *Auto* and Test Mark Type is set to *Tracking Mark*, then a tracking custom test mark fires when an input (part sensor) signal is detected on *Input Bit 0* (*IN0*) after encoder pulses satisfy the specified Sensor Distance. If the *Test Mark* pushbutton on the FH head is pressed, Tracker makes a static (Index) mark. *Tracking Mark* allows you to load a custom test mark file, disconnect the fiber link cable and computer, and then automatically mark a Tracking file each time a part moves through the marking field.

Note: The *Tracking Mark* option appears dimmed when WinMark Pro senses it is communicating with an FH or Fenix Index marking head.

Custom test mark file size is limited by the marking head's non-volatile memory. For example, the default test pattern fills approximately one-half of the available memory. When a custom test mark file is too large to download, WinMark displays "*The selected mark drawing does not fit in Test Mark Memory*". If this happens, try the following options to reduce file size:

A Reduce the Off Vector Resolution of mark objects in the drawing. Resolutions of 50–100 DPI are sufficient to obtain a quality mark.

B Reduce the Resolution of mark objects. 300 DPI is adequate for most marking. For graphics with only horizontal or vertical straight lines (not diagonals), 100 DPI is sufficient.

C Refine the mark. Use stroke text instead of TrueType fonts, mark fewer words, or simplify line art.

Note: Because it is a downloaded file stored in memory, a custom test mark file cannot perform serialization, real-time date and time coding, input/output, or other automation functions as it could when run on a computer using WinMark Pro.

Note: Creation of a custom test mark file can be done offline, but the marking head must be connected and communicating to WinMark Pro when setting custom test mark options.

To setup and download a custom test mark, perform the following steps:

1 Open an existing (or create a new) mark file.

2 Under *Tools*, click Custom Test Mark…. The *Test Mark Control* dialog opens.

3 Set the appropriate Test Mark Mode, *Manual* or *Auto*, and the Test Mark Type, *Index* or *Tracking*.

Note: The default Test Mark Mode is *Manual*. To enable *Auto* Test Mark Mode you must select the *Auto* option each time the *Test Mark Control* dialog box opens.

4 Click *OK* to download the currently active WinMark file to FH or Fenix memory.

5 After file transfer is complete, a dialog box confirms a successful download.

6 If the data transmission fails, click *OK* to acknowledge the failure and then repeat steps 2–5. If the Custom Test Mark… option appears dimmed after a data transmission failure, then under *Help*, click About Synrad WinMark… and then click *Head Info* to reset communications. After closing the dialog boxes, return to the *Tools* menu. The Custom Test Mark… option should be available.

Although the marking head can contain only one custom test mark at a time, a new file can be downloaded at any time. A copy of the factory-installed test pattern (*FactoryTestMark1.mkh*) is provided in the *Program Files\WinMark* folder so that the default test mark file can be reloaded into FH memory as required for lens testing or mark centering.

It is possible for the FH Series marking head or Fenix Laser Marker to set or clear FH/Fenix outputs (OUT4–OUT7) when marking a custom test mark. This output capability is only possible when using SetDigitalState commands in WinMark Pro's Event Builder. Event Builder functions are available during the following stages of the Automation loop:

On Before Mark Session—On Before Mark Piece—On After Mark Piece—On After Mark Session

Remember that in WinMark Pro's Automation loop, On Before Mark Session and On After Mark Session events run only once before/after a marking session while On Before Mark Piece and On After Mark Piece events run before/after each individual piece. In a custom test mark application however, each individual mark is considered a mark session meaning that On Before Mark Session / On After Mark Session events are also run for each piece that is marked. It is not possible for a custom test mark to perform non-Event Builder output functions (Set Digital Before Session, Set Digital Before Mark, or Set Digital After Mark) or to perform any type of input automation. Input signals are sensed in only three case: (1) for the Fast Acting Safety Interlock (FASI) function where *Input Bit 3* (*IN3*) must be active; (2) for Index marks where *IN0*, the part sensor input, is expected when Test Mark Mode is set to *Auto*; or (3) for Tracking where *IN0* is the part sensor input and *IN1*/*IN2* are position inputs from the rotary encoder.

⚠ Warning
serious personal injury

Marking head or Fiber Link Controller Card (FLCC) outputs may be set or cleared during the custom test mark download process when the mark file contains Event Builder automation.

The user must take steps to assure that this behavior does not compromise the integrity of the control system by causing unintended machine operation.

Virtual Laser Marking...

Underlay a high-resolution scanned image of the part to be marked on the current drawing. Use this feature to precisely align the mark to your part.

Synrad WinMark ActiveX Control Help

Open an online Help window describing how to use WinMark Pro's ActiveMark™ (ActiveX®) control.

Window

Window commands (Figure 16-16) allow you to perform functions such as switching between or re-arranging multiple drawing windows.

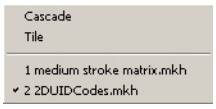


Figure 16-16 Window menu

Cascade

Overlap all open drawing windows.

Tile

Arrange all open drawing windows side-by-side.

"current active file"

Select any open drawing in the list as the currently active drawing.

Tip: You can also toggle the *Work Book* property on the View menu to show/hide *Work Book* tabs. Click a tab to quickly display each currently open drawing.

Help

Help commands (Figure 16-17) display program and resource information.

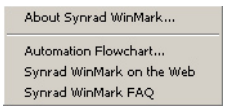


Figure 16-17 Help menu

About Synrad WinMark...

Display software version, copyright, and support information (Figure 16-18). You can also upgrade FH/Fenix marking head firmware (*Update F/W*) or check marking hardware/firmware status (*Head Info*).

Figure 16-18 About Synrad WinMark dialog box

Click the *Head Info* button (Figure 16-19) to see hardware/firmware information such as card type, head type, data transfer size, lens type, head firmware version, and FLCC firmware revision.

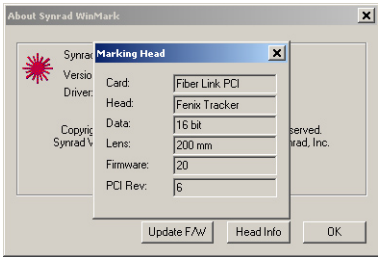


Figure 16-19 Marking Head dialog box

To upgrade the firmware in your FH Series marking head or Fenix Laser Marker (heads with firmware version 7 or greater only), perform the following steps:

1. Click the *Update F/W* button.
2. A dialog box opens asking you to initiate the firmware update. Press *OK*.

3 In the *Open* dialog box select the desired firmware file, *Update_xx.asc*, and press *Open*.

Note: Firmware files are copied to the *Program Files/WinMark* folder during installation.

4 After downloading the new firmware file, remove power from the marking head, wait a few seconds, and then power up the head to begin operation with the new firmware.

5 Check *Help/ About Synrad WinMark…/Head Info* to verify the new firmware version in memory. See your marking head's Operator's Manual for complete upgrade details.

Note: If, for any reason, you need to reverse the firmware upgrade, you can do so by pressing and holding the *Test Mark* button while applying power to the head. This action deletes the marking head's firmware upgrade and restores the original factory firmware.

Automation Flowchart...

Display a flowchart (Figure 16-20) indicating the order of events when using WinMark Pro automation features. See *Marking parts* in Chapter 10 for a detailed mark session flowchart.

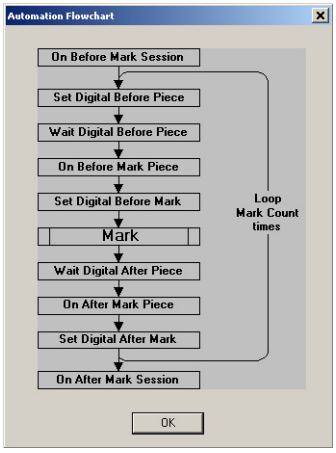


Figure 16-20 Automation flowchart

Synrad WinMark on the Web

Automatically link to the www.winmark.com web site through your browser. The new WinMark Pro web site contains a wealth of information including About WinMark Pro, Software Downloads (including sample files), Laser Marking FAQs, User's Guides and Operation Manuals, SYNRAD Marking Heads and Accessories, and WinMark Pro Support.

Synrad WinMark FAQ

Link directly to SYNRAD's online *Laser Marking FAQ*. This list of Frequently Asked Questions covers marking head and software topics in categories such as Product Upgrades, General Troubleshooting, WinMark Pro Automation, Tracking Operation, and ActiveX.

17 Drawing properties

The *Drawing properties* chapter includes subsections:

- Tracking tab
- Automation tab
- Marking tab
- Format tab
- All tab

All objects created or imported onto the *Drawing Canvas* have sets of properties associated with them. These properties allow you to define how each specific object is displayed or marked. In addition to objects you place on the *Drawing Canvas*, the *Drawing* object itself (the .mkh marking file) contains properties that allow you to apply global changes for tracking, marking output, or communication with automated parts handling equipment. Click on *Drawing* in the *Object List* or click the *Drawing Canvas* itself to select the *Drawing* object.

Drawing properties are classified by category in the *Property List* and include *Tracking*, *Automation*, *Marking*, *Format*, and *All* tabs. Click on the *All* tab to display all properties for the *Drawing* object.

Tracking tab

When WinMark Pro senses that it is communicating with an FH or Fenix Tracker it adds another tab, *Tracking*, to the *Property List*. Tracking properties (Figure 17-1) are used to set tracking parameters for part movement, speed, and direction.

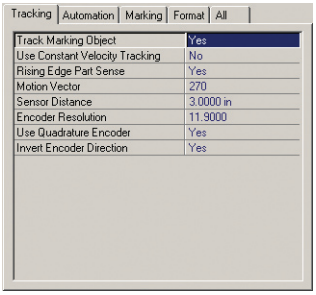


Figure 17-1 Tracking tab drawing properties

Track Marking Object

Select *Yes* to mark moving parts "on-the-fly" with the Tracker feature. Choose *No* to mark stationary objects in Index mode.

Use Constant Velocity Tracking

Set to *No* if your XY position data is generated by an external position encoder whose signal varies as part motion varies. If your products travel through the marking area at a constant speed, select *Yes* and enter a value for Product Line Speed to generate data at a fixed rate.

Rising Edge Part Sense

Set Rising Edge Part Sense based on how the part sensor output should trigger the mark. *Yes* triggers marking on the rising edge of the sensor's output waveform while selecting *No* triggers marking on the falling edge of the signal. Figure 17-2 illustrates a typical part sensor waveform.

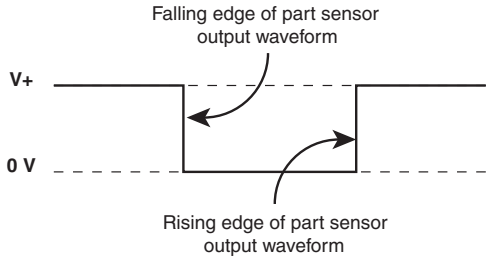


Figure 17-2 Part sensor output waveform

Motion Vector

Set the direction of part movement through the marking field. When looking at WinMark's *Drawing Canvas* (shown in Figure 17-3), 0° is part movement towards the left, 90° is movement towards the bottom, 180° is towards the right, and 270° is part movement towards the top of the *Drawing Canvas*.

When Track Marking Object is set to *Yes*, a motion arrow appears to the left of the *Drawing Canvas* showing the direction of the currently entered Motion Vector.

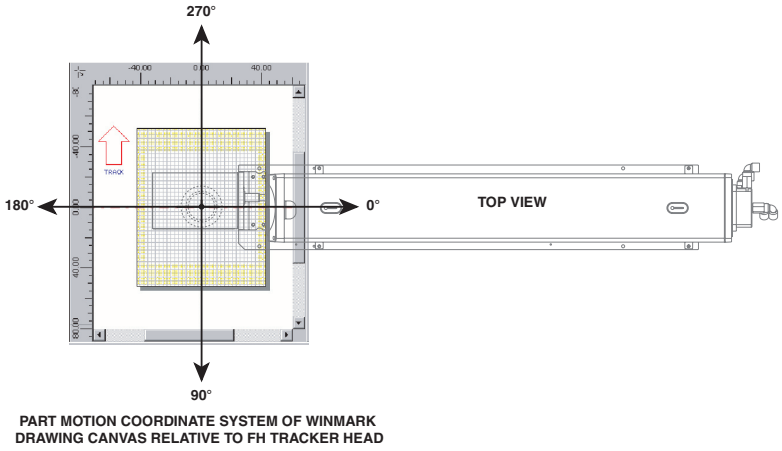


Figure 17-3 Motion Vector coordinates relative to Tracker

Tip: Part motions of 90° or 270° allow faster line speeds than 0° or 180° because part movement is through the long axis of the marking field, providing a larger *Tracking Window*.

Sensor Distance

Change Sensor Distance to move the location of the mark on the part in the axis of part motion. Units of measure for Sensor Distance are always entered in inches with a maximum value of 12.0 inches (305 mm).

The term, Sensor Distance, is not entirely accurate because the value entered is not related to the physical part sensor location but rather Sensor Distance is defined as ***the distance, at the moment of part sense, from the center of the Marking Window to the leading edge of the desired mark location***. Unlike other marking methods that are time-based, Tracker uses a distance-based marking scheme. This is because at the moment a part is sensed (on the rising or falling edge of the part sense signal), the FH/Fenix Tracker head begins counting rotary encoder pulses. When the number of encoder pulses counted by the head equals the Sensor Distance (calculated by the Encoder Resolution), marking begins. Because Tracker calculates marking vectors based on encoder pulses (distance), Tracker can continue to accurately mark an object even when it stops once the *Target Area* has fully entered the *Tracking Window*.

Note: Moving the location of the mark object on the *Drawing Canvas* in the axis of part motion has no affect on mark location when Tracking. Positioning mark objects further downstream on the *Drawing Canvas* only shortens the *Tracking Window*, which decreases tracking line speed. Change Sensor Distance to change the actual mark location on the part.

Figure 17-4 illustrates the Sensor Distance concept. In this case, the part sensor is placed upstream of the mark field (parts are sensed before they reach the center of the mark field) and the sensor is set to trigger on a rising edge transition.

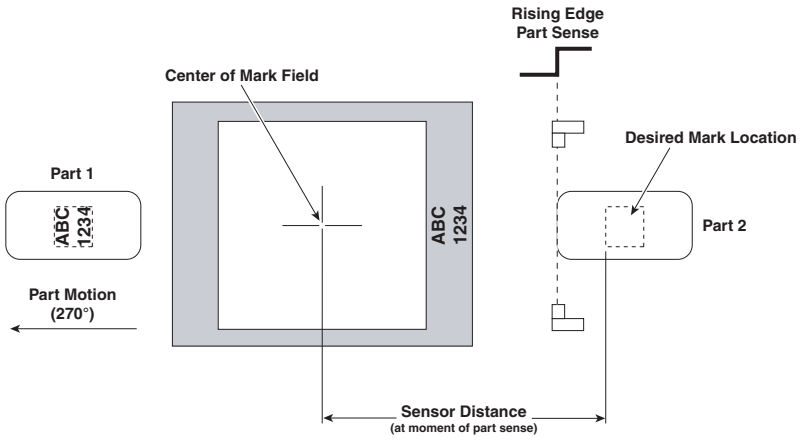


Figure 17-4 Upstream part sensor, rising edge trigger

Tip: Position the physical location of the part sensor so that the sensor does not trigger while the Fenix or FH head is marking a part.

Maximum Tracker line speeds are obtained when mark objects are positioned near the upstream edge of WinMark's *Drawing Canvas* (to provide the largest possible *Tracking Window*) and the Motion Vector (on the *Tracking* tab) is set to 90° or 270°, which takes advantage of the rectangular lens field.

Each object in the drawing has an X-Y 'start mark' point, which is the starting beam position. Sensor Distance must be greater than the value of the X or Y 'start mark' coordinate, which is normally the object's top left corner (unless the object has been rotated). When the Tracking Motion Vector is 90° or 270°, the Sensor Distance must be greater than the Y 'start mark' coordinate. When the Motion Vector is 0° or 180° then Sensor Distance must be greater than the X 'start mark' coordinate. If the Sensor Distance minus the mark position on the *Drawing Canvas* is less than or equal to zero, this means that the part will be past the point of marking before the specified Sensor Distance is traversed. If this error occurs, move the part sensor further upstream. If line speed (cycle time) is not an issue, you can instead move the object's location on the *Drawing Canvas*. In Figure 17-5, the text object's top left coordinate value is X = –0.5, Y = 2.0 (inches). If the *Drawing's* Motion Vector is 270°, then enter a Sensor Distance greater than 2.0 inches.

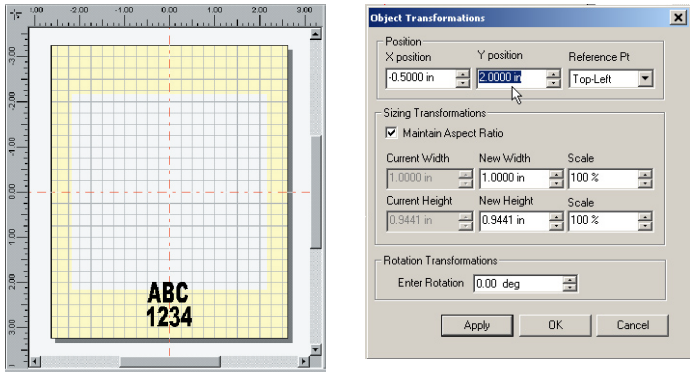


Figure 17-5 X-Y 'start mark' coordinates

Rotating an object changes the X-Y coordinates of its 'start mark' point, which may require you to reposition the mark object on the *Drawing Canvas* or change Sensor Distance (move the part sensor). Depending upon the Motion Vector selected, rotating an object 180° may allow object placement closer to the upstream edge of the *Drawing Canvas*, thus increasing overall line speed.

Encoder Resolution

Encoder Resolution is the number of encoder pulses received per millimeter of distance traveled by the parts conveyor. Measurement units for Encoder Resolution are fixed as pulses/mm even when WinMark is configured to display measurement units in inches or centimeters.

Consult the "Tracker" section in the *FH Series Marking Head Operator's Manual* or the *Fenix Laser Marker Operator's Manual* for detailed information on calculating and verifying the proper Encoder Resolution value.

Use Quadrature Encoder

Choose *Yes* to specify a quadrature encoder input where 'A' and 'B' phases generate direction and velocity information. If *No*, the expected input is a velocity signal from a unidirectional encoder. A unidirectional input is permissible in applications where the product transport does not reverse direction during marking.

Invert Encoder Direction

Invert Encoder Direction allows you to invert phasing of the encoder's quadrature (bi-directional) output signal so that the actual direction of part movement through the marking field is correctly sensed as "forward". This feature eliminates the need to physically change input field wiring on the marking head.

Product Line Speed

When Use Constant Velocity Tracking is *Yes*, enter a Product Line Speed value (in units of inches per second) equal to the actual part velocity. Adjust Product Line Speed as required to fine-tune tracking to actual part motion.

Automation tab

Automation events for the *Drawing* (Figure 17-6) occur at various stages during the mark process as illustrated in the Automation Flowchart (refer back to Figure 16-20, Automation flowchart). Certain events can execute predefined Event Builder commands that perform various automation functions while other events exert input/output (I/O) control through marking head or FLCC input and output bits. These events establish interface protocols between WinMark Pro and your parts handling equipment and are specific to each individual drawing. Refer back to Chapter 10, *Marking parts*, for a detailed flowchart illustrating the marking automation sequence.

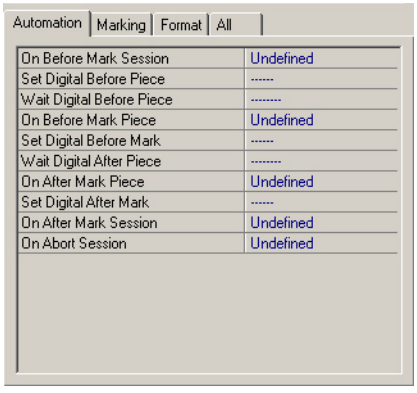


Figure 17-6 Automation tab Drawing events

On Before Mark Session

Define events you wish to trigger only once prior to the start of part marking in the current mark session. Click the ellipsis (…) to open the *Event Builder* dialog box and display a list of definable events. See Chapter 19, *Event Builder commands*, for descriptions of available commands.

Set Digital Before Piece

Set a specific digital output state on any or all output bits prior to each piece being marked. For example, your parts handling equipment may require that a specific output bit is set or cleared indicating the laser is off before staging a new part. See Chapter 12, *Automation basics*, for available outputs.

Wait Digital Before Piece

Wait for a specific digital input state to be read on any or all input bits before program execution continues. For example, your parts handling equipment may set or clear a specific input bit to indicate a new part is properly positioned. See Chapter 12, *Automation basics*, for available inputs.

Tip: Typically, *Input Bit 0* (*IN0*) and the Wait Digital Before Piece event are used to provide 'start mark' signals to WinMark Pro. When *IN0* is specified as *Set*, WinMark Pro waits until the specified input state is met—*Set* equals a logic high (1) state; *Clear* equals a logic low (0) state. WinMark continues executing the automation loop only after the specified input state is logically True.

On Before Mark Piece

Define events you wish to trigger once before each piece is marked. Click the ellipsis (…) to open the *Event Builder* dialog box and display a list of definable events. See Chapter 19, *Event Builder commands*, for descriptions of available commands.

Set Digital Before Mark

Set a specific digital output state on any or all output bits immediately before each piece is marked. For example, your parts handling equipment may require that a specific output bit is set or cleared to indicate that the laser is on and marking. See Chapter 12, *Automation basics*, for available outputs.

Important Note: WinMark Pro fires the laser immediately after the Set Digital Before Mark command is executed.

Wait Digital After Piece

Wait for a specific digital input state to be read on any or all input bits before program execution continues. For example, your parts handling equipment may set or clear a specific input bit to indicate a new part is being positioned. See Chapter 12, *Automation basics*, for available inputs.

Tip: Typically, *Input Bit 0* (*IN0*) and the Wait Digital After Piece event are used to provide the 'end mark' signal to WinMark Pro. When *IN0* is specified as *Clear*, WinMark Pro waits until the specified input state is met—*Set* equals a logic high (1) state; *Clear* equals a logic low (0) state. WinMark continues executing the automation loop only after the specified input state is logically True.

Tip: To minimize throughput delays, use Set Digital Before Mark to send a "laser on" signal to your automation controller. A signal at this step means that WinMark Pro has processed the 'start mark' signal. Your controller can then set the desired 'end mark' input state to ensure that there is no delay at the Wait Digital After Piece step once marking is complete.

⚠ Warning
serious personal injury

When the Wait Digital Before Piece Automation event is used to provide a 'start mark' signal, then Wait Digital After Piece must be used to verify that the 'start mark' signal has changed state. Failure to do so may cause unintended laser firing as WinMark Pro loops back through to the start of the automation sequence.

On After Mark Piece

Define events you wish to trigger once after each piece is marked. Click the ellipsis (…) to open the *Event Builder* dialog box and display a list of definable events. See Chapter 19, *Event Builder commands*, for descriptions of available commands.

Set Digital After Mark

Set a specific digital output state on any or all output bits immediately after each piece is marked. For example, your parts handling equipment may require that a specific output bit is set or cleared to indicate that the laser is off before staging a new part into position. See Chapter 12, *Automation basics*, for available outputs.

On After Mark Session

Define events you wish to trigger only once after all marking in the current session is complete. Click the ellipsis (…) to open the *Event Builder* dialog box and display a list of definable events. See Chapter 19, *Event Builder commands*, for descriptions of available commands.

Tip: A mark session ends when the number of parts marked equals the specified Mark Count value. In automated systems where Mark Count is zero (0) and marking proceeds continuously as directed by 'start mark' inputs, the mark session does not end. This means that On After Mark Session events are not executed.

On Abort Session

Define events you wish to trigger only once if the mark session is aborted or halted by the user (by pressing the ESC key for example). Click the ellipsis (…) to open the *Event Builder* dialog box and display a list of definable events. See Chapter 19, *Event Builder commands*, for descriptions of available commands.

Marking tab

Marking properties for the *Drawing* (Figure 17-7) perform global drawing transformations such as positioning or scaling your drawing on the part to be marked. Drawing transformations only affect actual marking output; the *Drawing Canvas* display does not change.

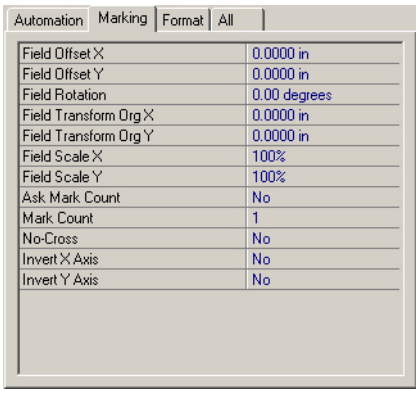


Figure 17-7 Marking tab Drawing properties

Field Offset X

Enter the amount of drawing offset required in the X-direction when marking. The drawing is offset from the transformation origin point specified by the Field Transform Org X and Field Transform Org Y properties.

Field Offset Y

Enter the amount of drawing offset required in the Y-direction when marking. The drawing is offset from the transformation origin point specified by the Field Transform Org X and Field Transform Org Y properties.

Field Rotation

Enter the amount of drawing rotation required around the transformation origin when marking. Field Rotation is accurate to two decimal places (0.01). The drawing is rotated around the transformation origin point specified by the Field Transform Org X and Field Transform Org Y properties.

Field Transform Org X

Specify the required transformation X-origin required when offsetting, rotating, or scaling a drawing. The default X origin point is 0.0000 inches.

Field Transform Org Y

Specify the required transformation Y-origin required when offsetting, rotating or scaling a drawing. The default Y origin point is 0.0000 inches.

Field Scale X

Specify the percentage of X-axis field scaling around the transformation origin.

Field Scale Y

Specify the percentage of Y-axis field scaling around the transformation origin.

Note: If Field Scale X and Field Scale Y are set to different values, the non-proportional field scaling causes mark distortion.

Ask Mark Count

If set to *Yes*, the *Piece Count* dialog box appears before each mark session begins. This dialog prompts the operator to enter the total number of marks required during the current mark session.

Mark Count

Specify the number of marks required during the mark session. If zero (0) is entered, marking continues until terminated by the ESC key or by event automation.

Important Note: When Mark Count is set to zero (0), you must configure WinMark Pro's automation and I/O features to wait for a 'start mark' signal from parts handling equipment so that marking continues only when an indication is received that another piece is in position and is ready to mark.

No Cross

Select *Yes* to prevent the beam from crossing a previously marked polyline. Use this feature to minimize hot spots on low threshold materials in situations where polylines intersect or cross each other.

No Cross Gap

When No Cross is set to *Yes*, enter a No Cross Gap dimension. No Cross Gap specifies the total non-marking gap across the intersected polyline.

Invert X Axis

Invert marking output along the horizontal axis.

Invert Y Axis

Invert marking output along the vertical axis.

Format tab

Format properties for the *Drawing* (Figure 17-8) allow you to specify a new field size if the focusing lens on your marking head is changed. See Tables 17-1 and 17-2 for focusing lens information.

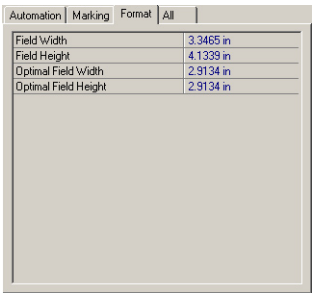


Figure 17-8 Format tab Drawing properties

Important Note: Field sizes on the *Format* tab do not update automatically when you choose a new lens in the *Lens Selection* dialog box. After selecting a new lens, you must always open a New *Drawing*.

Tip: If you accidentally create a mark file using the wrong lens for your marking application, simply select all objects and choose Copy from the Edit menu. Select the correct lens (Tools/General Settings… / Application Settings/Lens), open a New *Drawing*, and then Paste into the new file. WinMark Pro will paste all selected objects and their mark properties into the new drawing file.

Field Width

Maximum field width of the currently selected focusing lens.

Field Height

Maximum field height of the currently selected focusing lens.

Important Note: Field Width and Field Height dimensions (indicated by the yellow field on the *Drawing Canvas*) are lens specific and should not be changed. Changing the default mark field dimensions causes improper scaling of mark objects and, if changed non-proportionally, will produce distorted marks.

Optimal Field Width

Nominal field width of the currently specified focusing lens.

Optimal Field Height

Nominal field height of the currently specified focusing lens.

Table 17-1 and Table 17-2 list FH Series/Fenix and SH/DH focusing lenses, nominal and maximum field sizes for each lens, typical working distance, focused spot size and depth of field dimensions.

Table 17-1 FH Series/Fenix lens specifications

Lens Focal Length	Nominal Field H×W, mm	Max. Field H×W, mm	Working Distance[1] typical, mm	Spot Size (1/e²), µm	Depth of Field, mm
370 mm	198 × 198	241.0 × 297.0	350 ±5	540	± 10
200 mm	110 × 110	134.0 × 165.0	190 ±3	290	± 2.5
125 mm	74 × 74	85.7 × 105.6	128 ±2	180	± 1.5
125HP[2]	66 × 66	80.4 × 99.0	126 ±2	180	± 1.5
80 mm	27 × 27	33.5 × 41.2	74 ±1	116	± 0.4

1 for lenses with 19 mm (0.75") high lens mounts. The typical working distance is marked on each lens mount. Consult your marking head's final test report for the actual working distance.

2 125 mm lens for use with lasers 40 W and higher.

Table 17-2 SH/DH lens information

Lens Focal Length	Nominal Field H×W, mm	Max. Field H×W, mm	Working Distance[3] typical, mm	Spot Size (1/e²), µm	Depth of Field, mm
370 mm	200 × 200	200 × 200	455	530	± 10
200 mm	110 × 110	110 × 110	240	290	± 2.5
125 mm	70 × 70	70 × 70	148	180	± 1.5
69 mm	38 × 38	38 × 38	70	100	± 0.5

3 The exact distance is marked on the focusing lens mount because the actual working distance may vary from lens to lens within ±10 mm.

All tab

All applicable drawing properties from *Tracking*, *Automation*, *Marking*, and *Format* tabs are automatically included on this single tab (see Figure 17-9).

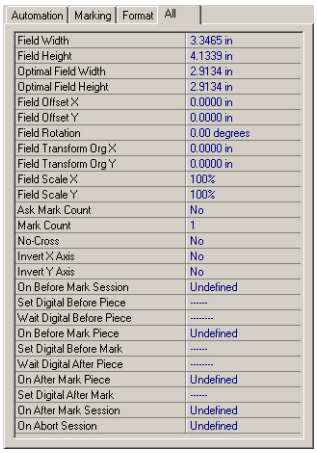

Automation
Marking
Format
All
Field Width 3.3465 in
Field Height 4.1339 in
Optimal Field Width 2.9134 in
Optimal Field Height 2.9134 in
Field Offset X 0.0000 in
Field Offset Y 0.0000 in
Field Rotation 0.00 degrees
Field Transform Org X 0.0000 in
Field Transform Org Y 0.0000 in
Field Scale X 100%
Field Scale Y 100%
Ask Mark Count No
Mark Count 1
No-Cross No
Invert X Axis No
Invert Y Axis No
On Before Mark Session Undefined
Set Digital Before Piece ------
Wait Digital Before Piece --------
On Before Mark Piece Undefined
Set Digital Before Mark ------
Wait Digital After Piece --------
On After Mark Piece Undefined
Set Digital After Mark ------
On After Mark Session Undefined
On Abort Session Undefined

Figure 17-9 All tab Drawing properties

18 Object properties

The *Object properties* chapter includes subsections:

- Special tab
- Automation tab
- Marking tab
- Format tab
- All tab

All objects created or imported on the *Drawing Canvas* have sets of properties associated with them. These properties allow you to define how each specific object is displayed and marked. Select an object on the *Drawing Canvas* or highlight an object name in the *Object List* to review or modify its properties. Object properties are classified by category in the *Property List* and include *Special*, *Automation*, *Marking*, *Format*, and *All* tabs.

Special tab

Special object properties (Figure 18-1) apply to objects contained within the drawing.

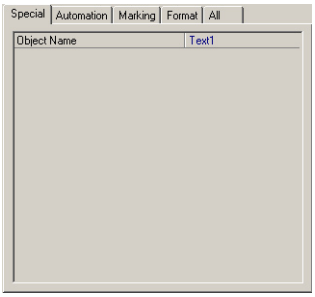


Figure 18-1 Special tab object properties

Object Name

Enter a specific descriptive name for each object created or imported onto the *Drawing Canvas*. Instead of using the default title "Text1", enter a name that better describes the object such as "PartNumber" or "Date Code String".

Automation tab

Automation object properties (Figure 18-2) apply only to text, 1D, and 2D bar code objects contained within the drawing. These properties help automate the serialization or coding of parts.

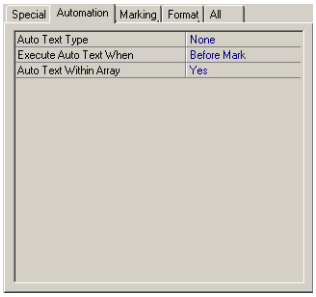


Figure 18-2 Automation tab object properties

Auto Text Type

Select the automated text input source from among *None*, *Serial Number Text*, *Date Code Text*, *Disk File Text* (Sequential or Fixed), *Serial Port Text*, *User Entry Text*, or *From Another Object*. Your choice of automated input determines which associated properties appear on the *Automation* tab.

Note: Some automated text and barcode captions are updated on the display when the object is selected or deselected. Dates, Times, and *From Another Object* sources are updated. Serial numbers are updated if Save Last Serial is *Yes*. Serial port and disk file text is not updated until the source is read during the mark sequence.

None

No automation is performed on the text object. The Text Caption is marked as displayed in the *Text Caption Editor* dialog box.

Serial Number Text

Serialize numerals, alphabetic characters, or alphanumeric strings entered as the Text Caption. When Serial Number Text is selected, the following serialization properties appear on the *Automation* tab: Serial Number Delta, Serial Number Mask, Serial Piece Step, Hexadecimal Count, Ask Starting Serial, Save Last Serial, Save Serial Name, and Current Saved Value.

Serial Number Delta

Choose a value with which to increment (positive values) or decrement (negative values) the serial number string before, or after, each mark.

Note: Only alphanumeric characters are serialized. Non-alphanumeric characters ("@", "*", etc.) remain as entered in the initial text string.

Serial Number Mask

Specify which alphanumeric characters in the serial number string should be serialized (the default setting) or fixed. From the drop-down list, select a placeholder position and then choose *Serialize* or *Fixed*.

Tip: Non-alphanumeric characters are ignored by the serialization protocol so that characters such as dashes (–), colons (:), etc need not be *Fixed* by Serial Number Mask.

Serial Piece Step

Specify how many pieces (parts) to mark before the serial number is incremented or decremented.

Hexadecimal Count

Choose *Yes* to increment/decrement serial numbers using the hexadecimal (base 16) number system. Choose *No* to count in decimal (base 10).

Ask Starting Serial

When set to *Yes*, a dialog box opens at the beginning of each mark session prompting the operator to enter a starting serial number.

Save Last Serial

Restore the last serial number marked when the next mark session begins.

Save Serial Name

Create a unique name for storing the Save Last Serial number. The actual number is stored in the computer's *System Registry* and so is not accessible by programs other than WinMark Pro.

Important Note: If more than one mark file exists for serializing parts, each file must specify a different Save Serial Name to prevent serial values from being overwritten.

Current Saved Value

Contains the current Save Last Serial number value. When creating a mark file, the initial value entered should contain as many placeholder zeros as the maximum number you wish to serialize before rolling back over to zero. For example, if only one zero is entered for Current Saved Value, then mark one =1, mark two = 2..... mark nine = 9, mark ten = 0, and mark eleven = 1.

Note: When serializing text, the character string entered for Current Saved Value, not Text Caption, is used for marking.

Date Code Text

Mark standard or customized date and time formats. When Date Code Text is selected, the following date code properties appear on the *Automation* tab: Assign Date Format, Time Offset: Days, Time Offset: Hours, and Time Offset: Minutes.

Assign Date Format

Click the ellipsis (…) to select or define specific date code formats. From the *Date Format Customization* dialog box, select from among American Standard, European Standard, Week-Year, or Julian formats or click the *Add Format* button to create an unlimited number of custom date formats. When coding by shift, the *Define Shifts* button allows you to customize shift start and shift end hours as well as shift code characters.

Note: Beginning with version 4.1.0.4368, custom date formats referenced in a mark file are saved in the mark file (instead of just in the development computer's *System Registry*). This prevents an "undefined" format error from occurring when files with Custom Date Formats are loaded on different computers.

To create a custom date format:

1. In the *Date Format Customization* dialog box (Figure 18-3) click the *Add Format* button.

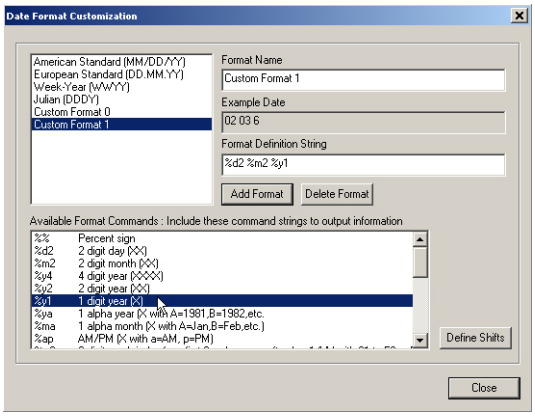


Figure 18-3 Date Format Customization dialog box

2. If desired, type a descriptive name in the *Format Name* box.
3. Browse the *Available Format Commands* box for available date or time formats.
4. Double-click the desired format commands or click in the *Format Definition String* box and type in commands to create a custom format string.

Tip: The percentage sign (%) is used by WinMark Pro to indicate a date format command. To add a percentage sign to your date format, type %%.

5 When done, click the *Close* button.

Tip: The *Example Date* box shows how the *Format Definition String* will look when marked. You can add spaces, dashes, or other fixed text within the string by typing characters between definitions in the *Format Definition String* box.

Time Offset: Days

Specify a positive or negative offset from the current day when printing expiration dates.

Time Offset: Hours

Specify a positive or negative offset from the current hour when printing expiration dates or fine-tuning shift code changes.

Time Offset: Minutes

Specify a positive or negative offset from the current minute when printing expiration dates or fine-tuning shift code changes.

Disk File Text (Seq)

Automatically open, read, and mark a line of text from a multiline disk file. Each time the specified file is read, the next (sequential) line of text in the file is assigned as the text object's Text Caption. When Disk File Text (Sequential) is selected, the following disk file properties appear on the *Automation* tab: Text Source Filename and Auto Text File Wrap.

Note: When using Disk File Text input, each text string must end with a Carriage Return (CR) character to indicate "end of data".

Text Source Filename

Specify a path and filename pointing to the location of the text source to read.

Auto Text File Wrap

If set to *Yes*, WinMark Pro reads each item in a sequential disk file and after the last item is read, wraps back around to the beginning item. Use Mark Count (on the *Marking* tab) to specify the number of iterations through the file. Because one line of text is read on each pass, each line in the disk file must end with a Carriage Return (CR) to provide the "end of data" delimiter.

Serial Port Text

Automatically open, read, and mark a text string from your computer's serial communications port. Each time the specified communications port is read, serial port data is assigned to the text object's Text Caption. When Serial Port Text is selected, the following serial port properties appear on the *Automation* tab: Text From Com Port, Com Port Timeout, and Close Comport After Read.

Note: When using Serial Port Text input, the text string must end with a Carriage Return (CR) character to specify "end of data". For multiline text strings, use a Line Feed (LF) character to force a line wrap and then end the last line with a Carriage Return.

Text From Com Port

Specify which communications port on your computer contains the incoming serial text string.

Click the Text From Com Port property and then click the down arrow to see a list of available com ports detected on your computer.

Click the ellipsis (…) to open the *Serial Port Settings* dialog. This dialog allows you to view and modify serial port settings (Figure 18-4) as required by your serial communications network. WinMark Pro v4 supports serial baud rates between 2400–115200 baud.

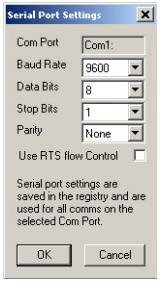


Figure 18-4 Serial Port Settings dialog box

Important Note: Prior to version 4.0.0.4051, WinMark Pro's serial communications rate was fixed at 9600 baud.

Com Port Timeout

Set the communications port timeout period in milliseconds. The mark will fail if information is not available at the port within the specified time. A timeout value of *–1* forces WinMark Pro to wait indefinitely until a message is received or the ESC key cancels the mark session.

Close Comport After Read

The default (and recommended) setting is *No*, which means the computer's com port is opened during the initial com port read and remains open throughout the mark session. Choosing *Yes* forces the com port to close after each read operation in cases where subsequent serial data must be ignored. The com port is reopened when the next serial port read is commanded.

User Entry Text

Prompt the operator to manually enter a text string for each piece in a mark session. When User Entry Text is selected, the following user entry property appears on the *Automation* tab: Text User Prompt.

Text User Prompt

Create a customized dialog box prompting the operator to enter the required mark data.

Disk File Text (Fixed)

Automatically open, read, and mark a line of text from a predefined disk file. Each time the specified file is read, the (fixed) line of text in the file is assigned as the text object's Text Caption. When Disk File Text (Fixed) is selected, the following disk file property appears on the *Automation* tab: Text Source Filename.

Note: When using Disk File Text input, the text string must end with a Carriage Return (CR) character to indicate "end of data".

Text Source Filename

Specify a path and filename pointing to the location of the text source to read.

Tip: Although only one line of text is read for each mark, the specified file can be updated between marks by external processes.

From Another Object

Assign a text caption for the currently selected Text, Barcode, or 2D Barcode object from another Text, Barcode, or 2D Barcode object. Use From Another Object to ensure that variable text data shared between two objects remains synchronized throughout a mark session. When From Another

Object is selected, the following object reference property appears on the *Automation* tab: Assign From.

Note: WinMark Pro will not link to reference objects in an array when the referenced object's Auto Text Within Array property is *Yes*. This occurs because each element in the array of reference objects is serialized and marked independently before the selected object or array of objects is marked. If these conditions are present, WinMark Pro will prevent the mark session from opening and display an error dialog.

Assign From

Copy Text Caption data from the referenced Text, Barcode, or 2D Barcode object. Use the Assign From property to link variable data between two or more text-based entities.

Tip: If the referenced object's Execute Auto Text When *Before Mark/After Mark* property is changed, WinMark Pro automatically changes the mark order of selected and referenced objects as required to ensure that Text Caption data remains synchronized. These automatic changes to mark order are recorded in the *Mark Log* at the beginning of a mark session. Where mark order is important, as in some Tracker applications, select the referenced object and choose the appropriate Execute Auto Text When *Before Mark/After Mark* property value to force the desired mark order.

Execute Auto Text When

Set the desired automation action to occur either before or after the piece is marked. For example, if your starting serial number value is "*000*" and Execute Auto Text When is *Before Mark*, serialization occurs before the mark and the first value marked is "001".

Note: When an object's Auto Text Type is set to Date Code Text, the Execute Auto Text When property is disabled; date code data is always read from the computer's system clock immediately before each mark.

Note: When the selected object's Auto Text Type property is set to: From Object, the Execute Auto Text When property is hidden and is replaced by the From Entity property. This occurs because the selected object derives its variable Text Caption data solely from the referenced object; it has no independent serialization capability.

Auto Text Within Array

Enable the desired automation action within any array elements that exist. Choose *Yes* to mark each array element with a different value. Choose *No* to mark all array elements with the same initial value.

Marking tab

Marking object properties (Figure 18-5) apply to objects contained within the drawing and specify how a particular object is laser marked. Most properties are common to all objects however; some properties apply only to specific types of objects such as text objects or bar code objects.

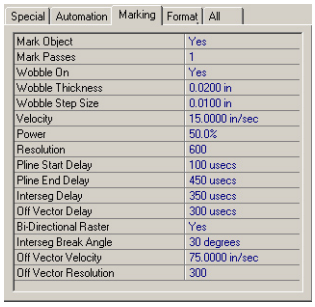


Figure 18-5 Marking tab object properties

Mark Object

Choose *Yes* to mark the current selection set. When *No* is selected, marking of currently selected objects is disabled. Although marking properties are hidden when marking is disabled, property values are retained by WinMark Pro.

Tip: The Boolean Mark Object property is often used in conjunction with MatchDigitalState automation so that a *True* input condition causes the specified object or group to mark. Objects/groups are not marked (Mark Object set to *No*) if the corresponding input condition is *False*.

Tip: It is often useful to have marking instructions or reminders placed on the *Drawing Canvas* in view of the equipment operator. Simply enter the desired instructional text, then size and position the text object. With only the text object selected, set Mark Object to *No*. Although visible, this text object will not mark and does not affect mark cycle times.

Mark Passes

Specify the number of passes required to mark the selected object. In some cases, several low-power passes are preferable to a single high-power pass.

Wobble properties

Wobble properties allow you to mark thick lines or strokes by "wobbling" the laser beam back and forth. Wobble lines are essentially thickened by marking shorter lines at right angles to the intended line segment. Line width is determined by the value specified for Wobble Thickness and line "resolution" is determined by Wobble Step Size. Although wobble provides the ability to create wide vector strokes, there is a significant increase in cycle time. Wobble is available for polyline, circle, stroke text, and imported vector objects.

Note: WinMark Pro imports some vector graphic files as a "Polyline Set" that must be exploded (*Objects* menu / Explode) into individual polylines before being "wobbled".

Wobble On

Choose *Yes* to mark thick line strokes by "wobbling" the laser beam.

Wobble Thickness

Specify the "wobble" width of line segments.

Wobble Step Size

Enter a specific distance between "wobble" steps.

Tip: In most cases, entering a Wobble Step Size smaller than the spot size of the focused beam serves only to increase cycle time without producing a visible difference in mark quality.

Velocity

Specify a galvanometer velocity when marking the selected object. When choosing Velocity values near the upper or lower limits, delay property values may require adjustment to prevent hot spots or "tails".

Power

Enter a percentage of PWM duty cycle (power) required to mark the selected object. Beginning with WinMark versions 4.3.3.4528 and above, Power can change in 0.1% steps. This change in resolution from 1% to 0.1% steps is supported by FH Series heads running firmware version 17 or above.

Tip: A value of 70–80% usually equals the laser's specified output power rating.

Resolution

Enter a marking resolution for the selected object. Units of measure are in microvectors per inch, which is roughly analogous to the more common dots per inch (DPI) printing terminology.

Note: Imported bitmap images are now resized according to the Resolution setting specified on the *Tools/General Settings…/Object Defaults* tab. Previously, WinMark Pro increased or decreased bitmap image size corresponding to a fixed bitmap resolution of 600 DPI. WinMark Pro does not resample bitmap images but rather resizes them to preserve the image's total pixel count—the overall dimensions of a 96 DPI image imported into WinMark at 300 DPI are automatically resized to 32% of their original height and width.

Tip: In order to import a bitmap image at full scale, WinMark's Resolution setting must match the file's resolution. For example, to import a file saved at 225 DPI, first ensure that the Resolution property on the *Tools/General Settings…/Object Defaults* tab is set to 225.

Delay properties

Because galvanometers require a finite amount of time to overcome mirror inertia and then move the mirror into position, galvanometer movement tends to lag slightly behind WinMark's software commands. Delay properties help "fine tune" the marking of each object in your drawing file.

If delay related marking problems occur, adjust marking delays by performing the following steps:

1. Select an object or group of related objects on the *Drawing Canvas*.
2. In the *Property List*, click the *Marking* tab.
3. Scroll down to the delay section shown in Figure 18-6.

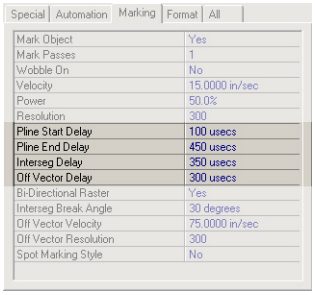


Figure 18-6 Delay settings

4. Write down your current delay settings for the problem objects.

5 Compare your delay settings with WinMark's default parameters:

Pline Start Delay – 100 µs

Pline End Delay – 450 µs

Interseg Delay – 350 µs

Off Vector Delay – 300 µs

6 Review Pline Start Delay, Pline End Delay, Interseg Delay, and Off Vector Delay descriptions on the following pages.

7 Make adjustments to the appropriate delays, ensuring that only one delay is adjusted at a time in order to avoid confusion and obtain the best trade-off between marking quality and marking speed.

Pline Start Delay

Use Pline Start Delay to minimize hot spots at the beginning of a polyline or the beginning of a series of polylines. See Figure 18-7 for illustrations of Pline Start Delay effects. Increase the delay to reduce dwell (hot spots) at the beginning of each polyline. If too large a delay is set, the beginning of a polyline may be shortened or shapes may not close properly. Pline Start Delay affects only the beginning of non-connected polylines.

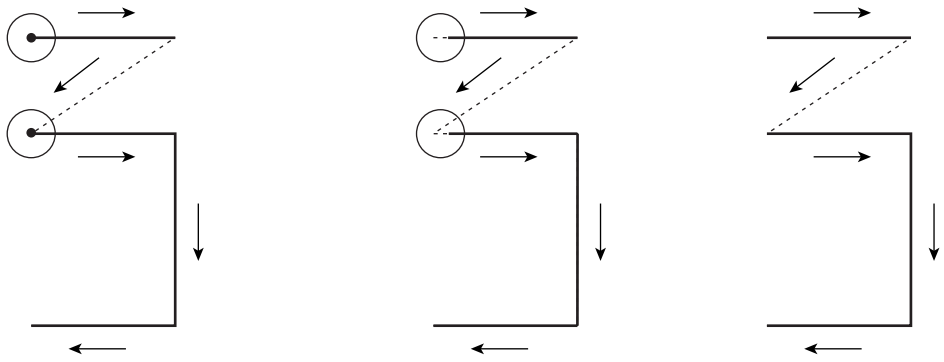


Figure 18-7 Pline Start Delay settings

Pline End Delay

Pline End Delay maintains "beam on" time at the end of a series of polylines to ensure that the current polyline is completed before moving to the next polyline vector. See Figure 18-8 for illustrations of Pline End Delay effects. Too much delay will create dwell, or hot spots, at the end of polylines; too little delay may cause short or incomplete polylines to be marked.

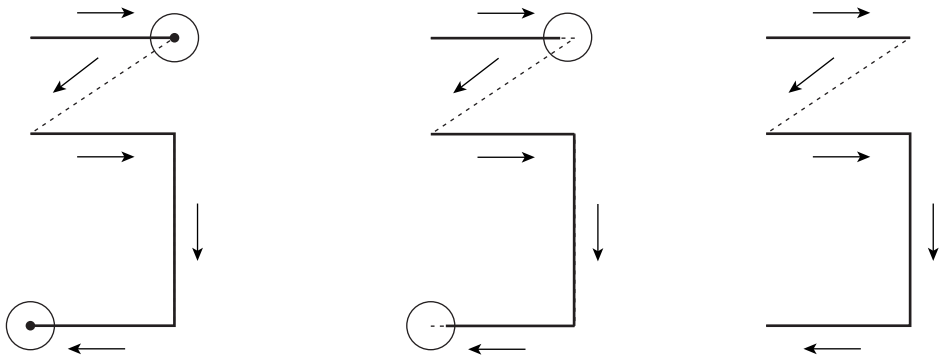


Figure 18-8 Pline End Delay settings

Interseg Delay

Interseg Delay sets a delay between marking connected polylines where the end point of the current polyline is the start point of the next polyline in the object. Interseg Delay increases the "sharpness" of points where polylines connect. See Figure 18-9 for illustrations of Interseg Delay effects. Too much delay creates dwell, or hot spots, at the end of polylines; too little delay causes corners to be rounded.

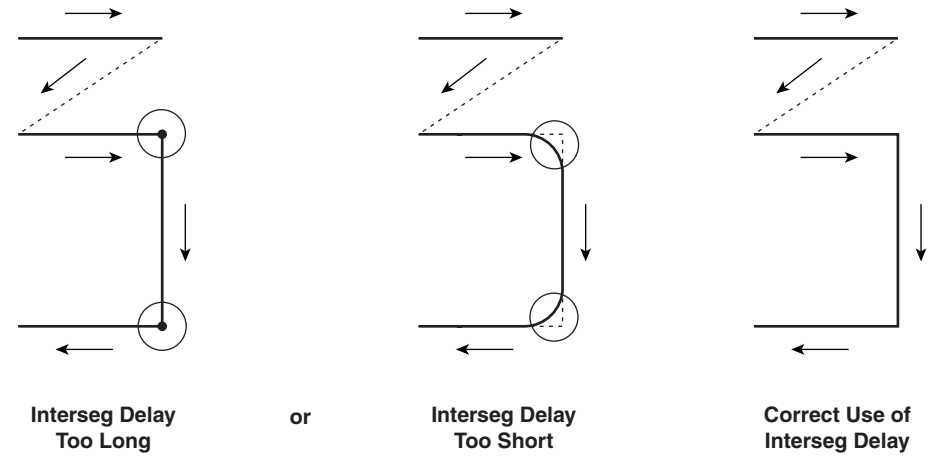


Figure 18-9 Interseg Delay settings

Off Vector Delay

Off Vector Delay sets a proportional delay during all laser off-vector moves to eliminate "tails" when moving between non-connected polylines. See Figure 18-10 for illustrations of Off Vector Delay effects. Too much delay will create a good mark, but marking throughput (cycle time) may be unacceptably slowed; too little delay may cause "tails", or leading marks at the beginning of polylines.

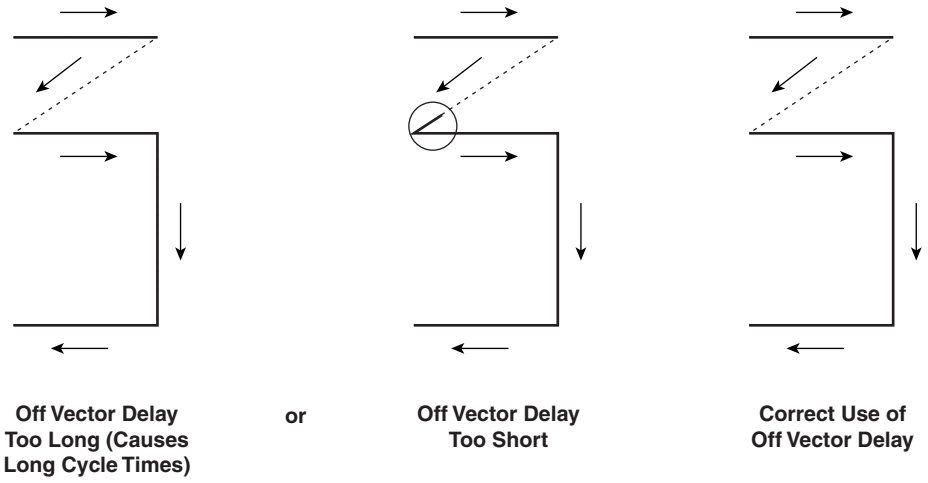


Figure 18-10 Off Vector Delay settings

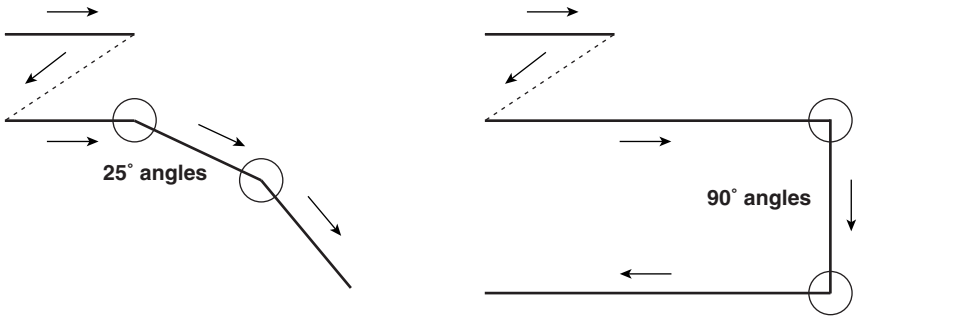
Bi-Directional Raster (not available for unfilled arc or polyline objects)

Set Bi-Directional Raster to *Yes* to reduce cycle time when marking a filled (raster) image. If set to *No*, laser marking of bitmap or filled images is done in only one scan orientation.

Note: When Bitmap Scan Type is set to *Continuous* (or when marking Grayscale images), Bi-Directional Raster is disabled; marking occurs in only one scan orientation although you can still specify a horizontal or vertical Raster Scan Direction.

Interseg Break Angle

Specifies a break angle between two connected polylines. No Interseg Delay is applied when the angle between two marked polylines is less than the specified angle. If the angle exceeds the Interseg Break Angle, then the Interseg Delay value is used. Figure 18-11 shows the results of marking two different angles using the default Interseg Break Angle of 30 degrees.



Angles are less than the specified Interseg Break Angle. No Interseg Delay is used.

Angles are greater than the specified Interseg Break Angle. Interseg Delay is used to prevent hot spots or rounded corners.

Figure 18-11 Interseg Break Angle settings

Off Vector Velocity

Set a non-marking velocity for galvanometers to travel between polyline endpoints. Generally, this number should be set close to the marking Velocity value.

Tip: Depending on the type of objects in the mark file, cycle times may be further reduced by increasing Off Vector Velocity towards its maximum value.

Off Vector Resolution

Set a non-marking resolution (microvectors-per-inch) when galvanometers travel between polyline endpoints. Generally, this number should be set close to the marking Resolution value.

Spot Marking Style

When set to *Yes*, the mark is made using individual spots instead of a continuous line. Spot spacing is proportional to the Resolution setting. Spot Marking Style is primarily used when marking glass to prevent fracturing or when marking very small vector-style 2D codes.

Spot Mark Duration

When Spot Marking Style is *Yes*, adjust the lasing duration in 1/10 of a millisecond (100 microsecond) intervals.

Format tab

Format object properties (Figure 18-12) apply to objects contained within the drawing. These properties set object positions and allow formatting of text and bar code characteristics. Most properties are common to all objects however; some properties apply only to specific types of objects.

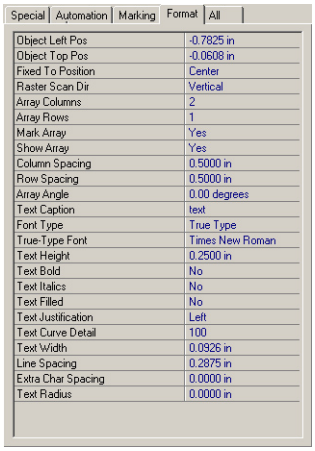


Figure 18-12 Format tab object properties

Object Left Pos

Set the left (X-coordinate) position for the selected object's bounding outline.

Object Top Pos

Set the top (Y-coordinate) position for the selected object's bounding outline.

Fixed To Position

Define a fixed reference point for positioning a text, bar code, or 2D code object when overall size is increased or decreased due to manual re-sizing, additional characters, etc. One of nine bounding outline handles is used as an anchor point; the other bounding outline points are free to move as re-sizing occurs.

Tip: When automatically serializing text strings containing lower case letters, the text string may shift up when lower case letters with descenders (g, j, p, q, s, y) are marked. This occurs because the Fixed To Position property defaults to *Center*. Set the text object's Fixed To Position property to *Top Left*, *Top Middle*, or *Top Right* for consistent vertical mark placement.

Raster Scan Direction

Specify horizontal or vertical raster scanning for filled objects or bitmaps.

Tip: When marking raster-scanned images such as imported graphics, filled TrueType text, or barcodes, one scan direction typically marks much faster than the other depending on the object's orientation. Optimize mark cycle time by ensuring that Bi-Directional Raster is *Yes* and then test mark each filled object using first a horizontal and then a vertical Raster Scan Direction.

Array properties

Array properties allow you to batch mark a group of parts. Select an object or objects to be marked as an array element then create an array by specifying rows, columns, and spacing. The mark location on pieces in the batch or tray of parts must all fit within the marking field of the selected focusing lens.

Array Columns

Enter the number of element columns in the array.

Array Rows

Enter the number of element rows in the array.

Mark Array

Choose *Yes* to mark the array. If *No*, only the original object, not the array, will mark.

Show Array

Select *Yes* to display the selected array or *No* to display only a single array element.

Column Spacing

Enter the desired spacing between element columns in the array.

Row Spacing

Enter the desired spacing between element rows in the array.

Array Angle

Enter a baseline angle for rotating the array.

Text format properties

Text properties (Figure 18-13) apply to text objects within the drawing. These properties provide typographical control over specific text characteristics.

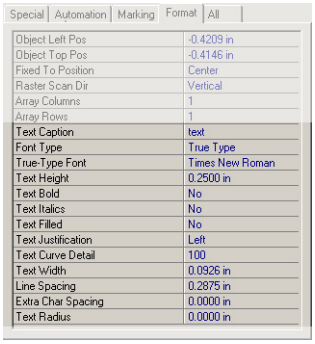


Figure 18-13 Format tab text properties

Text Caption

Enter a caption for the currently selected text object. If the text string displayed in the *Property List* is highlighted, typing will replace any current text. To edit, click once in the highlighted field to establish an insertion point or click the ellipsis (…) to enter multiline text using the *Text Caption Editor* dialog box.

Note: If Auto Text Type is set to other than *None*, the Text Caption string serves as only a placeholder and is replaced by the specified auto text during the mark session. For example, when serializing text, the character string entered as the Current Saved Value, not the Text Caption, is used for marking.

Note: Some automated text and barcode captions are updated on the display when the object is selected or deselected. Dates, times, and From Another Object sources are updated; serial numbers are updated if Save Last Serial is *Yes*. Serial port and disk file text is not updated until the source is read during the mark sequence.

Font Type

Select either TrueType or stroke fonts for text. Use stroke fonts for quick marks at high throughput speeds. Use TrueType fonts to match a logo, maintain a corporate identity, or create a certain "look".

Tip: Both stroke and TrueType fonts are marked as vector objects although TrueType fonts tend to have more complex path outlines. TrueType fonts are only marked using the slower raster-scanning technique when they are filled.

Stroke Font

Select one of WinMark Pro's twelve built-in stroke fonts. Fonts available include Bold, European, Gothic, LiteCom, Little, Sans Serif, Script, Simple, Trip, Tscr (Trip Script), Complex, and Simplex.

Tip: Complex and Simplex stroke fonts are proportional, or monospaced, fonts meaning that a lowercase "i" takes up as much letterspace as an uppercase "W". All other stroke fonts, and most TrueType fonts, are non-proportional. A proportional font is useful in situations where alphanumeric text is being serialized and any possible combination of letters or numbers must remain within a specific boundary on the part.

TrueType Font

Select a TrueType font for marking text. Any TrueType font installed in Window's *Fonts* folder is available for laser marking. When TrueType Font is selected, the following typographical properties appear on the *Format* tab: Text Bold, Text Italics, Text Filled, Text Outline Filled, Text Outline Power, and Text Curve Detail.

Note: WinMark Pro supports all TrueType (.TTF) fonts as well as the OpenType® (.TTF) fonts found on Windows 2000 and Windows XP operating systems. OpenType (.OTF) fonts are not currently supported. WinMark does not support Adobe® PostScript® (.PFM) fonts for Windows.

Text Bold

Select *Yes* to bold selected TrueType text.

Text Italics

Choose *Yes* to italicize selected TrueType text.

Text Filled

Select *Yes* to fill selected TrueType text. Choosing *Yes* causes WinMark Pro to fill text characters using a raster scanning method. Choosing *No* tells WinMark to mark text characters as vector outlines. When Text Filled is set to *Yes*, two additional properties, Text Outline Filled and Text Outline Power, appear.

Text Outline Filled

Choose *Yes* to outline the edges of raster-filled TrueType fonts. This vector outline tends to smooth the character shape after the raster-filling process.

Tip: When raster-filling text, try lower Power settings and/or lower Resolution values to prevent over-burning or distortion of the part substrate.

Text Outline Power

Set a power level for marking TrueType text outlines.

Tip: Typically, Text Outline Power is set higher than the Power level used for filling text.

Text Curve Detail

Specify a curve resolution for TrueType fonts. Higher Text Curve Detail values create smoother curves especially when marking large font sizes.

Text Height

Enter a base character height for text. Text Height is approximately the height of uppercase letters such as "M", "W", etc., although this height varies slightly depending upon the metrics of the chosen typeface. In general, Text Height for stroke fonts is padded to allow room for lower case descenders. This means that uppercase only stroke text displays and marks smaller than the value specified by Text Height.

Tip: To obtain accurate Text Height for your mark, use only uppercase letters and do the following: In the *Object Transformations* dialog box, check *Maintain Aspect Ratio*, set the *Current Height* value to the desired size, and then press the TAB key to scale text. Click *Apply* and then click *OK*.

Tip: Have you always specified point sizes when working with type? In typography there are 72 points per inch, so multiply your desired point size by 0.01389 to obtain a decimal value (in inches) for Text Height. For example, WinMark Pro's default Text Height is 0.25 inches, which corresponds to 18-point type.

Tip: To obtain the best legibility when marking text, the ratio of Text Height to beam spot size should be 7:1 or greater. For example, a 125 mm focusing lens has a spot size of 0.007" (0.18 mm). In this case, the minimum Text Height value should be 0.049 inches (1.26 mm).

Text Justification

Select left, right, or center justification for multiline text objects.

Text Width

Enter an average width (horizontal scale) for text characters. Enter zero (0) to return to WinMark's default font width.

Line Spacing

Enter line spacing (leading) between lines of multiline text. Enter zero (0) to return to WinMark's default spacing value.

Extra Char Spacing

Enter additional spacing (tracking) between text characters. Enter zero (0) to return to WinMark's default character spacing.

Text Radius

Enter a radius for marking text on a curve. Use Text Radius to fit text inside or outside an arc or circle. Positive values cause text to radius in a convex form; negative values cause text to radius in a concave form.

Polyline format properties

Polyline properties (see Figure 18-14) apply to polyline objects within the drawing. These properties give you the ability to customize polyline objects.

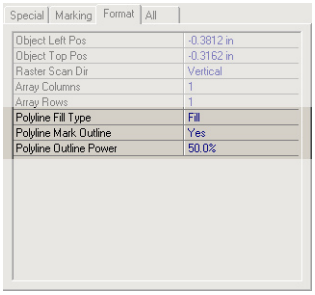


Figure 18-14 Format tab polyline properties

Polyline Fill Type

Fill the selected polyline object. Choosing *Fill* causes polylines to be filled by raster scanning. Choosing *No* simply marks polylines as vector outlines.

Polyline Mark Outline

When Polyline Fill Type is set to *Fill*, then choose *Yes* to outline polyline objects. This vector outline tends to smooth the polyline shape after raster filling.

Tip: When raster-filling objects, try a lower Power setting and/or a lower Resolution values to prevent overburning or distortion of the part substrate.

Polyline Outline Power

Set a power level for marking polyline outlines.

Tip: Typically, Polyline Outline Power is set higher than the Power level used for filling polyline objects.

Arc format properties

Arc properties (Figure 18-15) apply to arc, circular, and elliptical objects within the drawing. These properties give you the ability to create circles, ellipses, or arcs of varying sizes.

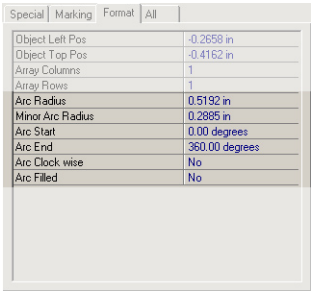


Figure 18-15 Format tab arc properties

Arc Radius

Enter an exact radius for a circular object.

Minor Arc Radius

For elliptical objects, enter a minor radius. For circles, enter the same value as the specified Arc Radius.

Arc Start

Use Arc Start to create an arc or semicircular object by specifying a starting angle for the object. Larger values move the starting position of the arc in a counterclockwise direction from the zero point. When Arc Start is used in conjunction with Arc End, the open arc segment can be made to point in any direction.

Arc End

Use Arc End to create an arc or semicircular object by specifying an ending angle for the object. Smaller values move the ending position of the arc in a clockwise direction from the zero point. When Arc End is used in conjunction with Arc Start, the open arc segment can be made to point in any direction.

Arc Clockwise

Choose *Yes* to mark the selected arc in a clockwise direction (in relation to WinMark Pro's *Drawing Canvas*). Choose *No* to mark in a counter clockwise direction. When the selected arc is not a closed object, changing Arc Clockwise from *No* to *Yes* inverts the arc segment and changes marking direction.

Arc Filled

Fill the selected arc. The Arc Filled property is only available for circular or elliptical objects. Open arcs cannot be filled.

Arc Mark Outline

Choose *Yes* to outline arc objects. This vector outline tends to smooth object edges after raster filling.

Tip: When raster-filling objects, try a lower Power setting and/or a lower Resolution values to prevent overburning or distortion of the part substrate.

Arc Outline Power

Set a power level for marking arc outlines.

Tip: Typically, Arc Outline Power is set higher than the Power level used for filling arc objects.

Bar code format properties

Bar code properties (Figure 18-16) apply to 1D bar code objects within the drawing. You can select specific bar code types and customize bar code properties.

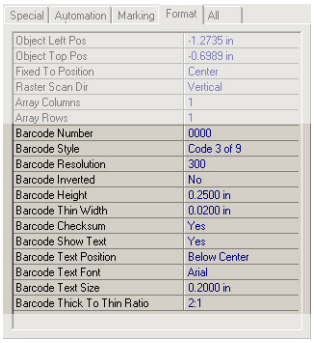


Figure 18-16 Format tab bar code properties

Barcode Number

Enter a caption for the currently selected bar code object. If the text string displayed in the *Property List* is highlighted, typing will replace any current text. To edit, click once in the highlighted field to establish an insertion point. For bar codes that support extended character sets, click the ellipsis (…) to enter multiline text using the *Text Caption Editor* dialog box. Barcodes supporting extended characters sets include PDF417, Extended Code 3 of 9, UCC-128, Code 128 (all 128 characters), and Extended Code 93.

Note: If Auto Text Type is set to other than *None*, the Barcode Number string serves as only a placeholder and is replaced by the specified auto text during the mark session. For example, when serializing text, the character string entered as the Current Saved Value, not Barcode Number, is used for marking.

Note: Some automated text and barcode captions are updated on the display when the object is selected or deselected. Dates, times, and From Another Object sources are updated; serial numbers are updated if Save Last Serial is *Yes*. Serial port and disk file text is not updated until the source is read during the mark sequence.

Tip: Bar codes not supported by WinMark Pro can often be created using third-party software. After creating the bar code, save it in a bitmap (.BMP) format, and then import the bitmap code directly into WinMark.

Barcode Style

Select from over twenty common bar code types. Table 18-1 shows supported 1D bar code formats.

Table 18-1 1D bar code formats

Supported Formats	Supported Formats
Codabar	HIBC Code 3 of 9
Code 3 of 9 (Code39)	Interleaved 2 of 5
Extended Code 3 of 9	MSI Plessey
Code 93	Portable Data File 417
Extended Code 93	Post Net 5, 9, 11 Digit
Code 128 (all 128 characters)	UCC-128
Code 128 Subset A	UPC-A 11 Digit
Code 128 Subset B	UPC-E 6 Digit, 0 Type Code
Code 128 Subset C	UPC-E 6 Digit, 1 Type Code
Ean/Jan 8 Digit	UPC-E 11 Digit
Ean/Jan 13 Digit	

Barcode Resolution

Define line density for the rasterized bar code. This value is used as the marking resolution.

Tip: In many cases, changing Barcode Resolution will alter the physical length of the bar code. To prevent this behavior when Barcode Thin Width is 0.02" or less, calculate the inverse or reciprocal of the Barcode Thin Width value, round it up, and then enter integer multiples of this number in order to determine the best Barcode Resolution for marking. For example, if your Barcode Thin Width value is 0.015" then the reciprocal is 1/0.015 or 66.66667. Round up this result to 67. To determine the best marking resolution, enter multiples of 67 (67, 134, 201, 268, etc.) as the Barcode Resolution and make test marks on your part. Using this method to constrain the length of the barcode will not always create the most compact code but it does ensure a consistent length as resolution is varied during testing.

Barcode Inverted

Create a contrasting "negative-image" of the bar code so that spaces between the bars are marked instead of the bars themselves. A "quiet zone" is added so that bar code readers can locate the first and last bars in the code. Barcode Inverted is used for marking dark materials that produce a lighter, contrasting mark.

Barcode Quiet Zone Size

If Barcode Inverted is *Yes*, you can choose to increase or decrease the length of the quiet zone.

Barcode Height

Set a specific height for the long bars in the code.

Barcode Thin Width

Specify a line width for thin bars in the code. Horizontal spacing is then proportionally adjusted throughout the code.

Tip: Depending on desired cycle time and material type, Barcode Thin Width should generally be set to mark a width equal to, or some multiple of, the spot size of the focusing lens. Assuming Barcode Resolution is optimized, a Barcode Thin Width value equal to the spot size means that only one pass is required to mark the line. Too large a value means that multiple slightly offset passes must be made to generate the wider line, potentially slowing cycle time. Marking too wide a line will also encroach on the white space between bars, causing verification problems.

Barcode Checksum

Choose *Yes* to append a Barcode Checksum (if applicable for the Barcode Style selected) to the Barcode Number value. Although the checksum is incorporated into the Barcode Number, it does not display as human-readable text (when Barcode Show Text is *Yes*).

Note: The Barcode Checksum property is suppressed for barcodes that do not support, or that always use, a checksum.

PDF 417 X/Y Ratio (for PDF 417 barcodes only)

Set the barcode cell X/Y (height to width) ratio by entering an integer value between 1 and 10. The default value is 4.

PDF 417 Security Level (for PDF 417 barcodes only)

Choose from one of nine levels of data security (error correction and detection). Each higher level of security adds additional overhead and physical size to the printed symbol. The default (lowest security level) is 0.

Barcode Show Text

Choose *Yes* to include human-readable text as part of the bar code object.

Barcode Text Position

If Barcode Show Text is *Yes*, select one of six locations for the placement of human-readable text.

Barcode Text Font

Select any TrueType font (installed in your *Fonts* folder) for the human-readable bar code text .

Tip: Human-readable bar code text always uses filled TrueType fonts. To shorten cycle times, set Barcode Show Text to *No*, create a new text object, and set Font Type to *Stroke*. If Barcode Number is a static value, simply enter this information into the text object's Text Caption. If Barcode Number is variable, use From Another Object and Assign From properties to link bar code and text objects so that data remains synchronized throughout a mark session.

Barcode Text Size

Enter a text size for the human-readable text included in the bar code.

Barcode Thick to Thin

Define a ratio for the width between thin bars/thin spaces and thick bars/thick spaces in the bar code. Based on the selected ratio (2:1, 2.5:1, or 3:1), horizontal spacing of thick bars/thick spaces is proportionally adjusted throughout the code.

2D code format properties

2D code properties (Figure 18-17) apply to 2D code objects within the drawing. You can choose to mark either Data Matrix or QR Code formats and customize 2D code properties.

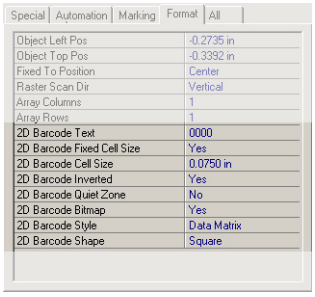


Figure 18-17 Format tab 2D code properties

2D Barcode Text

Enter single or multiline text up to 500 characters. If the text string displayed in the *Property List* is highlighted, typing will replace any current text. To edit, click once in the highlighted field to establish an insertion point. Click the ellipsis (…) to enter multiline text using the *Text Caption Editor* dialog box.

2D Barcode Fixed Cell Size

If 2D Barcode Fixed Cell Size is *No*, then cells are sized to fit the existing bar code bounding box. If set to *Yes*, then overall 2D bar code size is based on the number of cells and the individual cell size as specified by 2D Barcode Cell Size.

2D Barcode Cell Size

Specify the size of an individual 2D bar code cell when 2D Barcode Fixed Cell Size is *Yes*.

2D Barcode Inverted

Create a contrasting "negative-image" of the bar code. The spaces between the cells are marked instead of the cells themselves and a "quiet zone" is added so that 2D code readers can locate the outside edges of cells. This feature is used for marking dark materials that produce a lighter, contrasting mark.

2D Barcode Quiet Zone

Defaults to *Yes*, if 2D Barcode Inverted is selected. This property creates a one-cell (Data Matrix) or four-cell (QR Code) quiet zone around the code to aid readability.

2D Barcode Bitmap

Choose *Yes* to mark 2D code cells as standard bitmap (rastered) squares. Select *No* to mark cells as unfilled vector circles. At small code sizes, the unfilled vector circle is properly read as a "cell" with the primary advantage of substantially reduced cycle times. See Chapter 11, *Fast 2D codes*, for details on setting up and using this property.

2D Barcode Circle Radius

When 2D Barcode Bitmap is *No*, enter a value less than 100% to scale down vector circle (cell) size. Smaller diameter cells are useful for marking sensitive materials where overlapping vector strokes produce unacceptable mark quality.

2D Barcode Style

Select either *Data Matrix* or *QR Code* 2D code styles.

2D Barcode Shape

If 2D Barcode Style is *Data Matrix*, choose between square and rectangular code formats. Rectangular codes are useful for marking long thin parts but are limited to less than 98 digits or 72 alphanumeric characters. Entering characters beyond this limit forces the code to revert to a square.

Error Correction Mode

If 2D Barcode Style is *QR Code*, choose from one of four error correction modes. *High Density* creates the most compact code while *Ultra High Reliability* produces the highest accuracy when reading dirty or damaged codes.

Data Input Mode

If 2D Barcode Style is *QR Code*, choose *Automatic*, the default setting, or *Manual* data input. *Manual* input is used when multiple data types (numeric, alphanumeric, Kanji, and binary) are contained within a single data string.

For *Manual* data entry only, a character mode must precede the data string. Valid character modes are: N – Numeric; A – Alphanumeric; K - Kanji (Data Character String consists of shift JIS codes); and B - Binary (in Binary mode, the data is first preceded by the number of data characters represented as a four digit decimal number.)

To create a QR Code using *Manual* data input containing the numeric data "123456789012345", enter the following string in the 2D Barcode Text property: *N123456789012345*. To enter data in *Manual* mode using more than one character mode, place a comma between the first data string and the new character data mode. For example to enter the numeric data "12345" and the alphanumeric string "QRCode", enter the following string for 2D Barcode Text: *N12345,AQRCode*.

Spot format properties

Spot properties (Figure 18-18) apply to spot objects within the drawing.

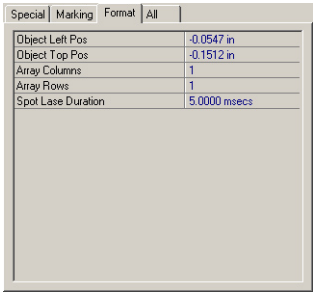


Figure 18-18 Format tab spot properties

Spot Lase Duration

Enter a lasing duration in milliseconds. Spot Lase Duration is used for marking spots or drilling thru holes in material.

Bitmap format properties

Bitmap properties (Figure 18-19) apply to imported bitmap images within the drawing.

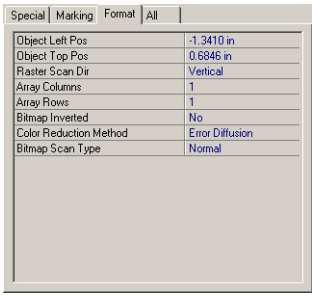


Figure 18-19 Format tab bitmap properties

Bitmap Inverted

When set to *Yes*, imported bitmap (raster) images are inverted to create a contrasting "negative-image" of the bitmap. As shown in Figure 18-20, dark areas are laser marked, while light areas are unmarked. Use this feature to mark dark materials that produce a lighter, contrasting mark.

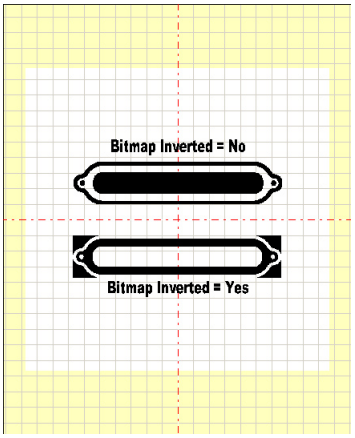


Figure 18-20 Normal and inverted bitmap images

Tip: For best results when marking bitmap (raster) images, set a Resolution value that corresponds to 100–150% of the spot size generated by the selected focusing lens. For example, a 200 mm lens with a spot size of 0.011" is capable of marking approximately 100 discrete dots per inch. Generally, set Resolution equal to 200–300 when using an 80 mm lens; 150–225 for a 125 mm lens; 100–150 for a 200 mm lens; and 50–75 when marking with a 370 mm lens.

Color Reduction Method

When importing color (as opposed to monochrome) bitmaps, choose a Color Reduction Method from among *Bayer Dithering*, *Error Diffusion*, *Halftone*, or *Grayscale*.

Bayer Dithering, *Error Diffusion*, and *Halftone* conversion methods covert color images into monochrome (1-bit) images suitable for laser marking. The differences in these three reduction methods are in the changes made to the size, shape, and spacing of pixels in the bitmap in order to create an illusion of shading since all pixels in a 1-bit image are either black (marked) or white (unmarked). Figure 18-21 shows an original 8-bit image and the resulting 1-bit image after being processed through WinMark Pro's color reduction algorithm.

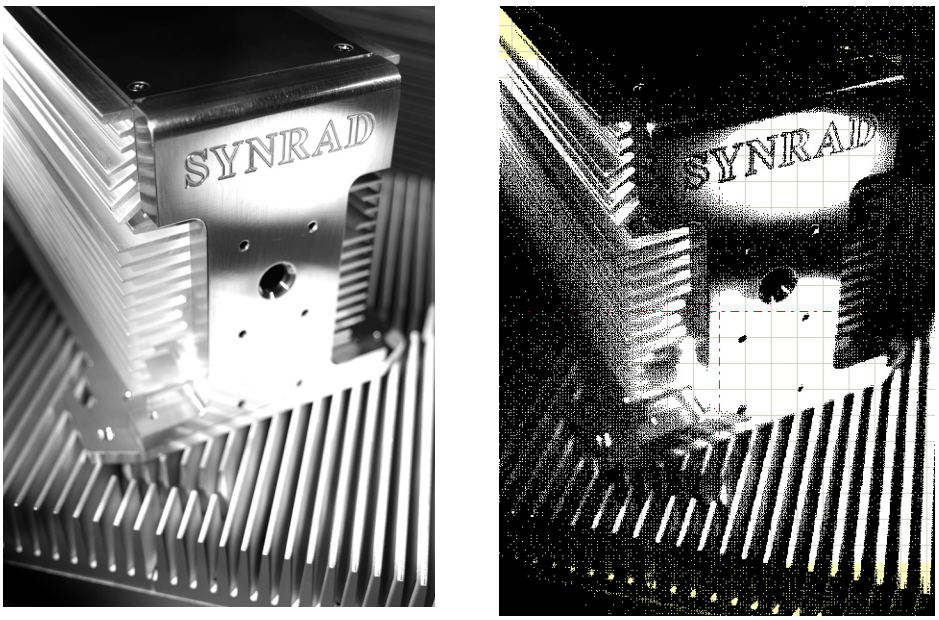


Figure 18-21 8-bit black & white image versus 1-bit Error Diffusion reduction

Note: Imported bitmap images are now resized according to the Resolution setting specified on the *Tools/General Settings…/Object Defaults* tab. Previously, WinMark Pro increased or decreased bitmap image size corresponding to a fixed bitmap resolution of 600 DPI. WinMark Pro does not resample bitmap images but rather resizes them to preserve the image's total pixel count—the overall dimensions of a 96 DPI image imported into WinMark at 300 DPI are automatically resized to 32% of the original height and width.

After changing Resolution and/or resizing the image, test mark the bitmap image. Use the Color Reduction Method property (on the *Format* tab) to change the color reduction algorithm as required. Any changes are performed on the original image, which is stored by WinMark during importation. Note that the screen display does not necessarily indicate the improved mark quality, so test marks are required.

Tip: Many times after choosing an initial Color Reduction Method and importing images into WinMark Pro, bitmap images are further resized or Resolution is altered. Because bitmap images are resolution-dependent (created for a particular display size and resolution), changing image size or resolution tends to degrade display and mark quality of the bitmap. If you must vary image size or resolution from the original settings, do so and then re-select a Color Reduction Method. In some cases, changing the reduction method produces better results.

Choose *Grayscale* reduction (Figure 18-22) to convert color bitmaps into 8-bit grayscale images (an 8-bit grayscale image provides 256 levels of gray ranging from 0–white to 255–black).

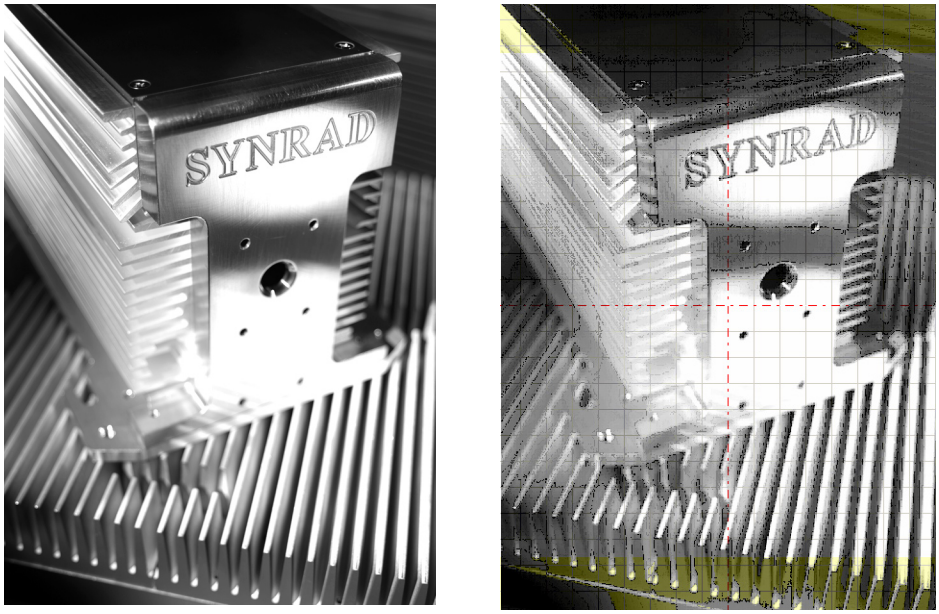


Figure 18-22 8-bit black & white image versus 8-bit Grayscale reduction

Important Note: FH Series Smart marking heads can mark monochrome (1-bit) bitmap images; however, FH Smart does not support grayscale marking.

Prior to FH firmware version 17 and WinMark Pro build 4.3.3.4528, laser power control was limited to 7-bit resolution because Power settings were limited to integer values (1% steps). This had a drastic effect on grayscale marks because it changed the apparent resolution of the image. For example: 7-bit resolution equals 128 levels (0–127); if the mark requires a Power setting of 30% to obtain the desired mark quality, image resolution becomes 30% of 127, which equals 38.1. With integer Power resolution, the original 8-bit (256 level) grayscale image effectively consists of only 38 levels of gray.

Beginning with version 17 and build 4528, WinMark Pro and FH Series heads support 8-bit Power resolution in 0.1% steps. Eight-bit Power resolution allows for 256 levels (0–255) of gray, effectively doubling the marked grayscale image resolution.

Tip: When marking grayscale images, higher Power levels provide higher image resolution (levels of gray). For best results, use a lower power laser and/or increase marking Velocity so that Power can be set in the range of 90–100%.

Tip: Before marking a grayscale image, create a matrix of filled squares as shown in Figure 18-23. By choosing various ranges for Power and Velocity, you can quickly determine settings that provide the best dynamic range for your particular image and substrate.

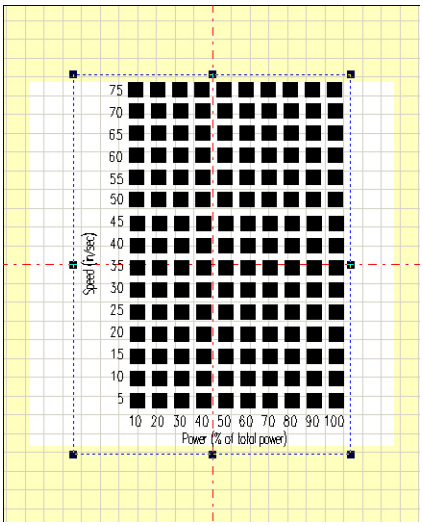


Figure 18-23 Sample Power/Velocity matrix

Bitmap Scan Type

Choosing *Normal* treats each scan line in a bitmap image as a sequence of microvectors that includes the corresponding Pline Start Delay, Pline End Delay, and Off Vector Delay times. Selecting *Continuous* forces each scan line to mark at a constant Velocity and Power, while laser power is switched on or off according to the bitmap data. Grayscale images are always marked using the *Continuous* scan method.

Note: Bi-Directional Raster is automatically disabled when Bitmap Scan Type is *Continuous*.

Note: If Bitmap Scan Type is *Continuous* or if the Color Reduction Method is set to *Grayscale* (which uses a *Continuous* scan method) then the product of multiplying Velocity and Resolution cannot exceed 33,000 units, which is the maximum command frequency of FH Series marking heads. In order to obtain an accurate representation of the image, WinMark Pro will display a dialog box and then automatically reset Velocity to the maximum value possible at the specified Resolution.

All tab

All applicable properties from *Special*, *Automation*, *Marking*, and *Format* tabs are included in this section (see Figure 18-24).

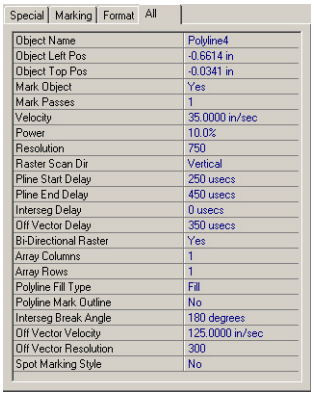


Figure 18-24 All tab object properties

19 Event Builder commands

Event Builder commands (Figure 19-1) are available at five different stages in the automation loop—(On Before Mark Session, On Before Mark Piece, On After Mark Piece, On After Mark Session, and On Abort Session). These commands perform automation functions during the marking process. In most cases, these automation events are assigned to a specific property of a specific target object.

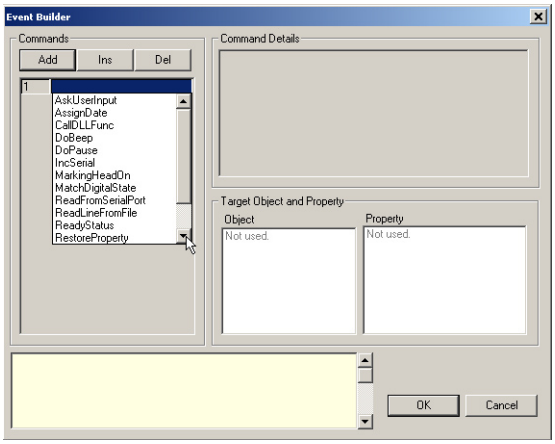


Figure 19-1 Selecting Event Builder commands

AskUserInput

Prompts the operator for keyboard input and assigns the resulting alphanumeric string to the specified target object and target property.

Tip: AskUserInput is often assigned to the Text Caption property of a text or 1D/2D barcode object.

AskUserInput Command Details:

User Prompt

Create a customized User Prompt for the *User Input* dialog box.

Use Default

If a new operator input string is not required for every mark, set Use Default to mark a predefined text string using From Current Property Value or From Default Text entries.

Default Text

Enter a default text string to mark if specific operator input is not required. The text entered here is marked when Use Default is set to From Default Text.

AssignDate

Allows you to create and assign a standard or customized date code format to a target object and target property (see Figure 19-2), typically a Text Caption property.

AssignDate Command Details:

Date Format

Click *Date Format* in the *Command Details* frame and then click the down arrow to select from a list of predefined standard or custom date formats or click the ellipsis (…) to open the *Date Format Customization* dialog and create new custom formats.

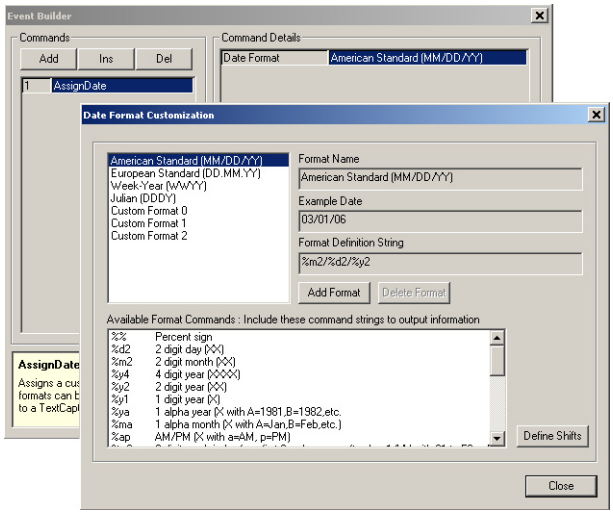


Figure 19-2 AssignDate command

To create a custom date format:

1. In the *Date Format Customization* dialog box click the *Add Format* button.
2. If desired, type a descriptive name in the *Format Name* box.
3. Browse the *Available Format Commands* box for available date or time formats.
4. Double-click the desired format commands or click in the *Format Definition String* box and type in commands to create a custom format string.
5. When done, click the *Close* button.

Tip: The *Example Date* box shows how the *Format Definition String* will look when marked. You can add spaces, dashes, or other text within the string by typing characters between definitions in the *Format Definition String* box. Because the percentage symbol (%) is used as a delimiter, you must type %% to add a single percentage symbol to the date code string you wish to mark.

Note: Beginning with WinMark versions after 4.1.1.4402, custom formats referenced in a mark file are now saved in the mark file (instead of just in the development computer's *System Registry*). This prevents an "undefined" format error from occurring when mark files with custom date formats are loaded on different computers.

CallDLLFunc

Calls a custom user-written Dynamic Link Library (DLL) function. The DLL can pass property and user arguments into WinMark Pro by specifying a *DLL Filename*, an export *Function Name*, and up to three user arguments.

CallDLLFunc Command Details:

DLL Filename

Enter the path and filename that points to the desired DLL file.

Function Name

Enter the DLL export function name (specified in the DLL's .def file).

User Arg #1

Enter the first argument, if any.

User Arg #2

Enter the second argument, if any.

User Arg #3

Enter the third argument, if any.

To activate a DLL from WinMark Pro, an event must be defined in the Event Builder via the *Automation* tab for the drawing. The following steps illustrate how to define a DLL using the example provided in the WinMark Pro *Samples* folder:

1. Open a New drawing and then place a text object on the *Drawing Canvas*.
2. Click the *Drawing Canvas* to select the *Drawing* object and then select the *Automation* tab from the *Property List*.
3. Select the marking automation event you want to define, such as On Before Mark Piece, and click the ellipsis (…).
4. In the *Event Builder* dialog box, click *Add* in the *Commands* frame.
5. In the field labeled *1* in the *Commands* frame, click the down arrow and select CallDLLFunc from the drop-down list.
6. In the *Command Details* frame, enter the DLL path and filename (*C:\Program Files\WinMark\Samples\Sample Dll\CommPortDll.dll*); the DLL export Function Name (*ReadSerialPort*), found in the *CommPortDll.def* file; and up to three user arguments (in this example, only two are used – *com1* for *User Arg#1* and *10000* for *User Arg #2*).
7. In the *Target Object and Property* frame, select a target object (*Text1*) and corresponding object property (Text Caption). Your ***Event Builder*** dialog should look similar to the one shown in Figure 19-3.

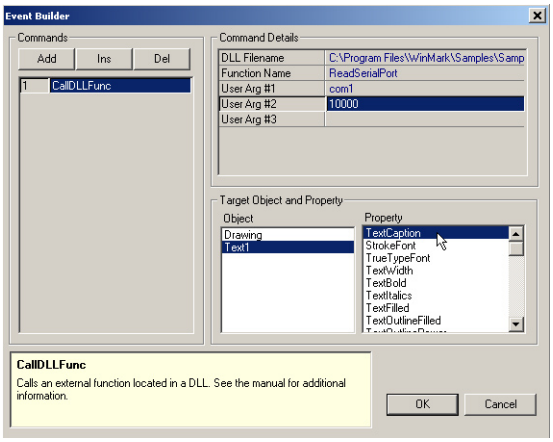


Figure 19-3 CallDLLFunc command

8 Click *OK*. When the drawing is marked, the DLL is executed.

This example opens *ComPortDll.dll* and calls the *SerialPortRead* function. If no data is read within the timeout period (10000 ms or 10 seconds), then the mark session aborts. If data is available on the designated serial port (COM1), then WinMark Pro reads the data, inserts a carriage return (CR) and line feed (LF) when a new line character is received, terminates the data input, and initiates the marking sequence when a "MARK" string is received. Serial data is stored in the *szPropBuffer* buffer and the *bBufferChanged* flag is set to a non-zero value. The flag tells WinMark that the received data should replace the current marking data.

To implement the DLL/WinMark interface, the DLL must use an argument structure with the following members:

```
struct _DLLStruct
    {
    char szUserArg1[64]; // argument 1
    char szUserArg2[64]; // argument 2
    char szUserArg3[64]; // argument 3
    char szPropBuffer[128]; // data buffer
    WORD bBufferChanged; // data change flag (object data changed in current drawing)
    void* pfnFuncCallback;
    void* pfnCallbackData;
    HWND hwndWinMark;
    };
```

The DLL's export functions must be explicitly defined in the .def file (for example: function1 @1, function2 @2, etc.).

DoBeep

Generates a beep from the computer's speaker.

Tip: This command might be used to provide an audible prompt for user input or a signal that marking is complete.

DoPause

Pauses program execution for the specified number of milliseconds.

DoPause Command Details:

Delay (msecs)

Enter a delay value in milliseconds. The default is 1000, which equals 1 second.

IncSerial

Increments or decrements alphanumeric characters in the currently selected text string by the specified delta value. Non-alphanumeric characters are ignored. This command is normally assigned to the Text Caption property.

Important Note: Unlike the serialization feature available from the Auto Text Type object property, Event Builder automation does not provide the ability to save the current serial value at the end of a mark session.

IncSerial Command Details:

Delta

Enter a positive integer value (to increment) or a negative integer value (to decrement) the serial count.

MarkingHeadOn

Defines an output signal (from the Fiber Link Controller Card only!) to parts handling equipment indicating a "No End of Mark Response" signal from a Fenix/FH Series marking head. Bits labeled IGNORE are not changed. A "No End of Mark Response" message is generated by WinMark Pro when the head fails to respond after mark data is sent. This can occur if the head loses power or the fiber optic communication link fails during a mark session. Output connections for the MarkingHeadOn event are made to the FLCC because marking head outputs are unavailable if the marking head is powered down. See Chapter 12, *Automation basics*, for a table showing available marking head and FLCC outputs.

MarkingHeadOn Command Details:

PCI Card ID

Enter a card ID corresponding to the ID of the card used for the input/output (I/O) connection. Typically this ID number corresponds to the ID of the card used for controlling the marking head (see Tools/General Settings…/Application Settings/IO Card Selection).

Output Bit 1
This output is available only from the Output mini-DIN connector on the FLCC.

Output Bit 3
This output is available only from the Output mini-DIN connector on the FLCC.

Output Bit 4
Corresponds to *OUT4* on Fenix/FH Series marking heads. This output is also available from the Output mini-DIN connector on the FLCC.

Output Bit 5
Corresponds to *OUT5* on Fenix/FH Series marking heads. This output is also available from the Output mini-DIN connector on the FLCC.

Output Bit 6
Corresponds to *OUT6* on Fenix/FH Series marking heads. This output is also available from the Output mini-DIN connector on the FLCC.

Output Bit 7
Corresponds to *OUT7* on Fenix/FH Series marking heads. This output is also available from the Output mini-DIN connector on the FLCC.

MatchDigitalState
Reads the input bit pattern of the marking head or the Fiber Link Controller Card (FLCC) and assigns the result of this Boolean test (Yes/True if all bits match; No/False if not) to the target property (see Figure 19-4). The input state of any bits labeled IGNORE is disregarded. The MatchDigitalState result is normally passed to Boolean properties such as Mark Object to include or exclude an object from marking. See Chapter 12, *Automation basics*, for a table showing available marking head and FLCC inputs.

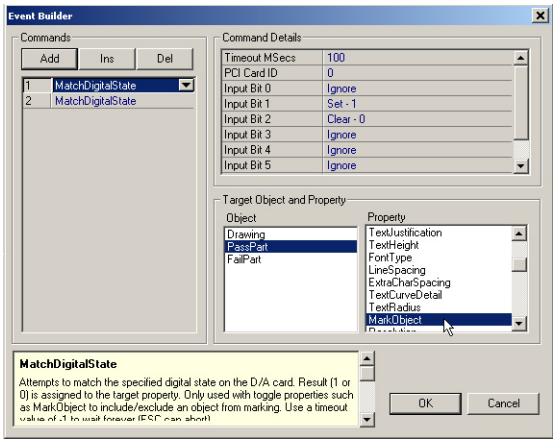


Figure 19-4 MatchDigitalState command

Tip: MatchDigitalState is useful for selecting one of a number of objects (or group of objects) to mark in cases where you have eight or fewer objects (or groups) to choose from during a single mark session. For example, if your production process randomly places different products on the marking conveyor asynchronously, you can configure WinMark Pro to mark one of eight different marks as determined by the state of marking head inputs. See Chapter 12, *Automation basics*, for a table showing available marking head and FLCC outputs.

MatchDigitalState Command Details:

Timeout MSecs
: Use a short timeout value (milliseconds or tens of milliseconds) to complete the input status scan immediately or enter a timeout value of *–1*, which forces WinMark to wait indefinitely for a valid match (or until the ESC key aborts the mark session).

PCI Card ID
: Enter a card ID corresponding to the ID of the card used for the input/output (I/O) connection. Typically this ID number corresponds to the ID of the card used for controlling the marking head (see Tools/General Settings…/Application Settings/IO Card Selection).

Input Bit 0
: Corresponds to *IN0* on Fenix/FH Series marking heads.

Input Bit 1
: Corresponds to *IN1* on Fenix/FH Series marking heads.

Input Bit 2
: Corresponds to *IN2* on Fenix/FH Series marking heads.

Input Bit 3
: Corresponds to *IN3* on Fenix/FH Series marking heads.

Input Bit 4
: Corresponds to *IN4* on FH Smart heads. On Fenix or FH Index/Tracker marking heads this input is available only from the Input mini-DIN connector on the FLCC.

Input Bit 5
: Corresponds to *IN5* on FH Smart heads. On Fenix or FH Index/Tracker marking heads this input is available only from the Input mini-DIN connector on the FLCC.

Input Bit 6
: Corresponds to *IN6* on FH Smart heads. Not available on Fenix or FH Index/Tracker heads.

Input Bit 7
: Corresponds to *IN7* on FH Smart heads. Not available on Fenix or FH Index/Tracker heads.

Follow the steps below to set up a demonstration file where only one of three objects marks on each pass through the automation loop, depending on the status of input bits *IN2* and *IN3*. MatchDigitalState works by evaluating the status, high or low, of the actual input bit(s) against the specified match values

and assigns a Boolean value of *True* or *False* to the target object and target property.

Configure MatchDigitalState automation as follows:

1. Open a new mark file and place three text objects on the *Drawing Canvas*. Give each text object a different caption so that they are uniquely identified.
2. Select the *Drawing* object (highlight *Drawing* in the *Property List)* and click the *Automation* tab.
3. On the Automation tab, click the ellipsis (…) to the right of On Before Mark Piece to open the *Event Builder* dialog.
4. Click the *Add* button to add a new command.
5. Click on the drop down arrow and select MatchDigitalState.
6. Set the *Timeout Msecs* value (in the *Command Details* frame) to *100*.
7. Enter a card ID number corresponding to the ID of the card used for the input/output (I/O) connection. Typically this ID corresponds to the ID of the card used for controlling the marking head (see Tools/General Settings…/Application Settings/IO Card Selection).
8. Set the *Input Bit 2* (*IN2*) state to 'Clear – 0' and set *Input Bit 3* (*IN3*) to 'Set – 1'. Leave all other bits set to 'Ignore'.
9. In the *Target Object and Property* frame, select the first of the three text objects as the target object, and select *Mark Object* as the target property.
10. Follow steps 4 through 7 to add a second MatchDigitalState command, but set *Input Bit 2* (*IN2*) to 'Set – 1', set '*Input Bit 3* (*IN3*) to 'Clear – 0', select the second text object as the target object, and choose *Mark Object* as the target property.
11. Follow steps 4 through 7 to add a third MatchDigitalState command, but set *Input Bit 2* (*IN2*) to 'Set – 1', set *Input Bit 3* (*IN3*) to 'Set – 1', select the third text object as the target object, and choose *Mark Object* as the target property. At this point, the Event Builder should look similar to the view shown in Figure 19-5 although your text object names may differ.

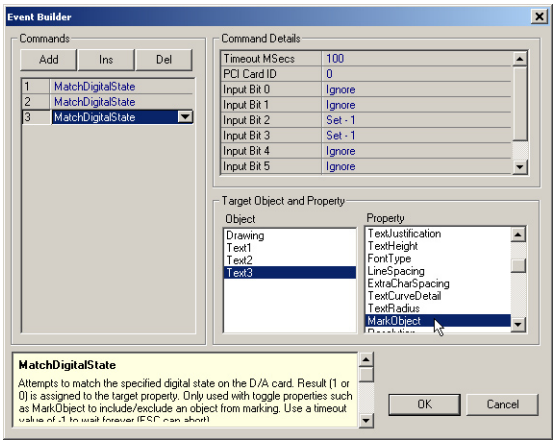


Figure 19-5 Event Builder setup for MatchDigitalState demo

12 Click *OK*, then save the mark file.

The *Timeout Msecs* property determines how many milliseconds WinMark Pro waits for input status to match the specified values before returning a value of *False* to the target object and property. WinMark Pro continuously scans the status of the input bits until the required bit pattern is matched or until the timeout value expires. As soon as the input pattern matches the specified input state, any remaining timeout period is ignored and WinMark Pro advances to the next step in the automation sequence.

Important Note: When multiple MatchDigitalState events are defined in the Event Builder, WinMark Pro begins scanning the list of MatchDigitalState events in succession. If the specified input state does not match, WinMark Pro waits the specified timeout period before continuing on to the next MatchDigitalState event in the list. For this reason, the cycle time of each mark will reflect the sum of the individual timeout periods specified for each non-matching input state.

The demo file described above is configured so that only one of the three MatchDigitalState commands will set its corresponding target object and property to *True*, provided that the state of the inputs are held steady during the succession of MatchDigitalState evaluations. This results in only one object being marked for each pass through the mark loop (see the *Automation Flowchart* found under the Help menu). No mark is made if the state of the inputs do not match one of the three MatchDigitalState input conditions.

Once your MatchDigitalState automation works as desired, you can add 'start mark' inputs and setup the mark file to execute the mark loop as many times as desired using the basic automation scheme described in Chapter 12, *Automation basics*.

ReadFromSerialPort

Reads a line of text from the specified serial port. The value read can be used to dynamically set the value of any target property. Since one text string is read from the serial port each time ReadFromSerialPort is called, you must end the line with a Carriage Return (CR) character to provide the "end of data" delimiter. For multiline text strings, use a Line Feed (LF) character to force a line wrap and then end the last line with a Carriage Return.

Tip: While the Auto Text Type property simply assigns incoming serial data to an object's Text Caption, Barcode Number, or 2D Barcode Text property, the Event Builder's ReadFromSerialPort event provides much greater flexibility. For example, you can assign serial port data to almost any *Marking* or *Format* property of the mark object or to a property of the *Drawing* itself.

ReadFromSerialPort Command Details:

COM Port

Specify a serial communications port on your computer, and then click the down arrow to see a list of available com ports detected on your computer. Click the ellipsis (…) to view and modify serial port settings (Figure 19-6) as required by your serial communications network. WinMark Pro versions 4 and above support serial transfer rates between 2400–115200 baud.

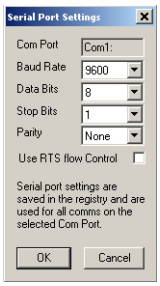


Figure 19-6 ReadFromSerialPort command

Timeout MSecs

Set a communications port timeout period in milliseconds. The mark will fail if information is not available at the port within the specified time. Entering a timeout value of *–1* forces WinMark Pro to wait indefinitely until a message is received or the ESC key cancels the mark session.

Close After Read

Use CloseAfterRead to determine whether the com port remains open or is closed after each serial port read. The default setting is *No*, which means the computer's com port is opened during the initial com port read and remains open throughout the mark session. Choosing *Yes* forces the com port to close after each read operation for cases where subsequent serial data must be ignored. The com port is reopened when the next serial port read is commanded.

ReadLineFromFile

Reads a line of text from a text (.txt) file and assign it to any target object or target property. One line of text is read each time ReadLineFromFile is called so you must end each text string with a Carriage Return (CR) to provide the "end of data" delimiter.

ReadLineFromFile Command Details:

Filename

Enter a path and filename that points to the desired text file on your computer.

ReadyStatus (Fenix Laser Marker only)

Defines an output signal (from either the Fenix Laser Marker or the Fiber Link Controller Card) to parts handling equipment indicating that Fenix did not mark due to a fault. Bits labeled IGNORE are not changed. This command queries Fenix for status after completion of a mark and sends the user-defined output signal if the marking head is not ready due to an error. See Chapter 12, *Automation basics*, for a table showing available marking head and FLCC outputs.

ReadyStatus Command Details:

PCI Card ID

Enter a card ID corresponding to the ID of the card used for the input/output (I/O) connection. Typically this ID number corresponds to the ID of the card used for controlling the marking head (see Tools/General Settings…/Application Settings/IO Card Selection).

Output Bit 1
This output is available only from the Output mini-DIN connector on the FLCC.

Output Bit 3
This output is available only from the Output mini-DIN connector on the FLCC.

Output Bit 4
Corresponds to *OUT4* on Fenix/FH Series marking heads.This output is also available on the Output mini-DIN connector on the FLCC.

Output Bit 5
Corresponds to *OUT5* on Fenix/FH Series marking heads.This output is also available on the Output mini-DIN connector on the FLCC.

Output Bit 6
Corresponds to *OUT6* on Fenix/FH Series marking heads.This output is also available on the Output mini-DIN connector on the FLCC.

Output Bit 7
Corresponds to *OUT7* on Fenix/FH Series marking heads.This output is not available on the FLCC.

RestoreProperty
Restores the specified target property's value from a disk file. The disk file name is your mark file name appended with a .dat extension and is always located in the *C:\Program Files\WinMark* folder. Refer to the SaveProperty event for more detail.

SaveProperty
Saves the specified target property's value to a disk file. The disk file name is your mark file name appended with a .dat extension and is always located in the *C:\Program Files\WinMark* folder. When using automation or serialization in your mark file, SaveProperty / RestoreProperty events are useful to ensure your mark file always close and reopen with a known or default set of property values.

Tip: Multiple SaveProperty and RestoreProperty events can be defined for a single drawing. Even though the data is stored in a single .dat file, a separate entry is created for each *Target Object* and corresponding *Target Property*.

SetDigitalState

Sets individual output bits on the marking head or Fiber Link Controller Card (FLCC) to the specified values. Bits labeled IGNORE are not changed. See Chapter 12, *Automation basics*, for a table showing available marking head and FLCC outputs.

SetDigitalState Command Details:

PCI Card ID

Enter a card ID corresponding to the ID of the card used for the input/output (I/O) connection. Typically this ID number corresponds to the ID of the card used for controlling the marking head (see Tools/General Settings…/Application Settings/IO Card Selection).

Output Bit 1

This output is available only from the Output mini-DIN connector on the FLCC.

Output Bit 3

This output is available only from the Output mini-DIN connector on the FLCC.

Output Bit 4

Corresponds to *OUT4* on Fenix/FH Series marking heads.This output is also available on the Output mini-DIN connector on the FLCC.

Output Bit 5

Corresponds to *OUT5* on Fenix/FH Series marking heads.This output is also available on the Output mini-DIN connector on the FLCC.

Output Bit 6

Corresponds to *OUT6* on Fenix/FH Series marking heads.This output is also available on the Output mini-DIN connector on the FLCC.

Output Bit 7

Corresponds to *OUT7* on Fenix/FH Series marking heads.This output is not available on the FLCC.

WaitDigitalState

Waits for a specific input pattern to be read on the marking head or Fiber Link Controller Card (FLCC) inputs. Program execution continues if the required state is met before the specified timeout interval; if not, the mark will fail. Input bits labeled IGNORE are disregarded. See Chapter 12, *Automation basics*, for a table showing available marking head and FLCC inputs.

WaitDigitalState Command Details:

Timeout MSecs

Use a short timeout value (milliseconds or tens of milliseconds) to complete the input status scan immediately or enter a timeout value of *–1*, which forces WinMark to wait indefinitely for a valid match (or until the ESC key aborts the mark session).

PCI Card ID

Enter a card ID corresponding to the ID of the card used for the input/output (I/O) connection. Typically this ID number corresponds to the ID of the card used for controlling the marking head (see Tools/General Settings…/Application Settings/IO Card Selection).

Input Bit 0

Corresponds to *IN0* on Fenix/FH Series marking heads.

Input Bit 1

Corresponds to *IN1* on Fenix/FH Series marking heads.

Input Bit 2

Corresponds to IN2 on Fenix/FH Series marking heads.

Input Bit 3

Corresponds to *IN3* on Fenix/FH Series marking heads.

Input Bit 4

Corresponds to *IN4* on FH Smart heads. On Fenix or FH Index/Tracker marking heads this input is available only from the Input mini-DIN connector on the FLCC.

Input Bit 5

Corresponds to *IN5* on FH Smart heads. On Fenix or FH Index/Tracker marking heads this input is available only from the Input mini-DIN connector on the FLCC.

Input Bit 6

Corresponds to *IN6* on FH Smart heads. Not available on Fenix or FH Index/Tracker heads.

Input Bit 7

Corresponds to *IN7* on FH Smart heads. Not available on Fenix or FH Index/Tracker heads.

20 ActiveMark technology

The *ActiveMark technology* chapter includes subsections:

- ActiveMark control overview
- ActiveMark control properties
- ActiveMark control methods

Important Note: When developing your automated control system using ActiveX control methods, do not place input automation into the mark file. Instead, perform any input automation functions within the ActiveX control program. By doing this, your control program remains in charge of pre- mark and post-mark automation and WinMark's MarkDrawing control method only maintains control of the operating system long enough to control laser on/off commands during the actual marking process. Performing input automation within the mark file may prevent the ActiveX control program from regaining control of the operating system if the specified input state does not occur.

ActiveMark control overview

For best results, use the WinMark Pro application to layout drawing elements and then load and automate the drawing via the ActiveMark™ (ActiveX/OLE) control to create custom marking solutions. The customized WinMark Pro ActiveX® control software usually does the following:

- Loads one or more WinMark Pro drawings into the control
- Manipulates drawing objects through property modification, position, or transformations
- Synchronizes with parts handling equipment
- Marks the drawing

In this *User Guide*, ActiveX control method descriptions are formatted as follows:

Control method/property description–

Control Method/Property Name and Format < Argument1, Argument2, … > RETURNED VALUE

Visual Basic/Visual C++ sample code–

```
Sample code line 1
Sample code line 2...
```

Most ActiveX control methods return a numerical or Boolean value to the calling program. Returned values are one of seven types:

BOOLEAN – returns a Boolean (1=True, 0=False) value to indicate if the command was successful.

FLOAT – returns a floating-point value.

INTEGER – returns an integer value.

LONG – returns a long integer value.

SHORT – returns a short integer value.

STRING – returns an alphanumeric string.

VOID – no value returned by control method.

Sample code

Sample Visual Basic® and Visual C++® code is provided in the *WinMark\Samples\VisualBasicSample* and *WinMark\Samples\VisualCPlusSample* folders respectively.

Additional VB code samples are available on the WinMark Pro CD or from the WinMark Pro web site at http://www.winmark.com/ActiveX_Sample_Files/activex_sample_files.htm.

Note: All Visual Basic code samples shown below assume that the WinMark Pro ActiveX control is named "mh". If you have chosen a different control name, the example code must be modified to reflect the new VB control name.

Important Note: We strongly recommend that all mark files run under ActiveX control use inches as the unit of measure. The calling program must calculate the conversion to inches from other units of measure. Unpredictable object placement or operation may result if the Show Units Type property is set to millimeters or centimeters.

Tip: In order to be recognized by the WinMark Pro OCX, ActiveX control methods and property names must be called using their internal program name. These internal names are written without spaces in English.

For example, to add a text object with an Object Name and Text Caption when the Application Language is Spanish, you would follow this convention:

```
mh.AddText "Objeto 1 Del Texto", 0, 0, "Accionada por Synrad!"
```

To change the Text Caption of an object (when the Application Language is German) and then command WinMark to mark the text using two Mark Passes, follow this convention:

```
mh.SetStringProp "Text1", "TextCaption", "Energie durch SYNRAD!"
mh.SetIntProp "Text1", "MarkPasses", 2
```

XY coordinate system

The XY object coordinates used in WinMark Pro are not Cartesian coordinates (refer back to Figure 4-4). On WinMark's *Drawing Canvas*, X-coordinate values are negative to the left of the origin; positive to the right while Y-coordinate values are negative above the origin and positive below.

ActiveMark control properties

DrawingIndex property

The DrawingIndex property is used to store multiple mark files in one instance of the WinMark Pro ActiveX control. Any number of drawings can be active, limited only by available computer memory.

The default value assigned to this property is zero (0).

Any operations performed on the control are performed on the current active drawing. To switch to another drawing, just assign a new number to the DrawingIndex control. Screen display of the control will not be updated until the Redraw method is called. The control property is formatted as:

DrawingIndex INTEGER

Visual Basic example:

This example opens and loads three mark files into memory. Each file is then sequentially activated and marked.

```
mh.DrawingIndex = 0
mh.LoadDrawing "test1.mkh"
mh.DrawingIndex = 1
mh.LoadDrawing "test2.mkh"
mh.DrawingIndex = 2
mh.LoadDrawing "test3.mkh"
mh.DrawingIndex = 0
mh.MarkDrawing
mh.DrawingIndex = 1
mh.MarkDrawing
mh.DrawingIndex = 2
mh.MarkDrawing
```

ProgressCtrlHandle property

The hWnd ProgressCtrlHandle property enables the display of a progress bar during lengthy WinMark Pro operations such as loading, saving, or marking a document. To use this property, create a Progress Bar control on the same form as your ActiveX control and assign ProgressCtrlHandle the window handle for your ActiveX control.

When you no longer want to display progress updates, assign 0 to the control. The control property is formatted as:

hWnd ProgressCtrlHandle

Visual Basic example:

```
' select the active drawing index
mh.DrawingIndex = 0
' assign the progress handle
mh.ProgressCtrlHandle = ctlProgressBar.hWnd
' load the drawing – a progress bar will now be displayed
mh.LoadDrawing "test.mkh"
mh.MarkDrawing
' no more progress displays
mh.ProgressCtrlHandle = 0
```

ActiveMark control methods

AboutBox method

The AboutBox control displays WinMark Pro's *About Synrad WinMark* dialog containing version and build information. The control method is formatted as:

AboutBox < none > BOOLEAN

Visual Basic example:

```
' display WinMark version info for operator
mh.AboutBox
mh.Redraw
```

Add2DBarCode method

The Add2DBarCode control adds a new 2D code object to the current drawing at the indicated coordinates using the current default object properties specified in the WinMark Pro application. The string entered as *Object Name* is assigned as the object name for the new 2D code. The control method is formatted as:

Add2DBarCode < ObjectName as String, X as Float, Y as Float, Text as String > BOOLEAN

Visual Basic example:

```
mh.Add2DBarCode "2DBarcode1", 0, 0, "AB241"
mh.Redraw
```

Tip: To change the 2D Barcode Style of a 2D barcode object, refer to the following chart:

2D Barcode Style	Property value
Data Matrix	0
QR Code	1

For example, if the default 2D Barcode Style is *Data Matrix*, add the following line of code to change the object labeled as *2D Barcode1* to a *QR Code* object:

```
mh.SetStringProp "2D Barcode1", "2DBarcodeStyle", "1"
```

Tip: When 2D Barcode Style is *Data Matrix*, you can choose a square or rectangular 2D Barcode Shape.

2D Barcode Shape	Property value
Square	0
Rectangular	1

For example, if the default 2D Barcode Shape is *Square*, add the following line of code to change the object labeled as *2D Barcode1* to a *Rectangular* shape:

```
mh.SetStringProp "2D Barcode1", "2DBarcodeShape", "1"
```

Tip: When 2D Barcode Style is *QR Code*, you can choose an Error Correction Mode.

Error Correction Mode	Property value
High Density	0
Standard Level	1
High Reliability	2
Ultra High Reliability	3

For example, to change Error Correction Mode to *Ultra High Reliability*, add the following line of code to modify the object labeled as *2D Barcode1*:

```
mh.SetStringProp "2D Barcode1", "ErrorCorrectionMode", "3"
```

Tip: When 2D Barcode Bitmap is *Yes*, WinMark Pro raster marks cells by filling them. Choose *No* to mark cells using an unfilled vector circle. Refer to Chapter 11, *Fast 2D codes*, for information on marking vector- or spot-style 2D codes.

2D Barcode Bitmap	Property value
No (mark vector circles)	0
Yes (mark filled squares)	1

For example, if the default 2D Barcode Bitmap property is *No*, add the following line of code to mark a Data Matrix or QR Code by raster-filling cells:

```
mh.SetStringProp "2D Barcode1", "2DBarcodeBitmap", "1"
```

AddBarCode method

The AddBarCode control adds a new bar code object to the current drawing at the indicated coordinates using the current default object properties specified in the WinMark Pro application. The string entered as *Object Name* is assigned as the object name for the new bar code. The control method is formatted as:

AddBarCode < ObjectName as String, X as Float, Y as Float, Number as String > BOOLEAN

Visual Basic example:

```
mh.AddBarCode "BarCode1", 0, 0, "12345"
mh.Redraw
```

Tip: To change the Barcode Style of a barcode object, refer to the following chart:

Barcode Style	Property value
UPC-A 11 Digit	0
UPC-E 11 Digit	1
Ean/Jan 13 Digit	2
Ean/Jan 8 Digit	3
Code 3 of 9 (Code39)	4
Extended Code 3 of 9	5
Interleaved 2 of 5	6
Code 128 (all 128 characters)	7
Codabar	8
Post Net 5, 9, 11 Digit	9
MSI Plessey	10

Barcode Style	Property value
Code 93	11
Extended Code 93	12
UCC-128	13
HIBC Code 3 of 9	14
Portable Data File 417	15
UPC-E 6 Digit, 0 Type Code	16
UPC-E 6 Digit, 1 Type Code	17
Code 128 Subset A	18
Code 128 Subset B	19
Code 128 Subset C	20

For example, if the default Barcode Style is *Code 3 of 9*, add the following line of code to change the object labeled as *Barcode1* to a *Code 128* (all 128 characters) code type:

```
mh.SetStringProp "Barcode1", "BarcodeStyle", "7"
```

Tip: When Barcode Show Text is *Yes*, choose a Barcode Text Position from among the following:

Barcode Text Position	Property value
Off*	0
Below Left	1
Below Center	2
Below Right	3
Above Left	4
Above Center	5
Above Right	6

* A Barcode Text Position property value of "0" is overridden by the state of Barcode Show Text (*No* corresponds to text Off; *Yes* corresponds to text On).

For example, if the default Barcode Text Position property is *Below Center* (2), add the following line of code to left-justify human-readable text above the code:

```
mh.SetStringProp "Barcode1", "BarcodeTextPosition", "4"
```

Tip: To specify a Barcode Thick to Thin Ratio, refer to the following chart:

Barcode Thick to Thin Ratio	Property value
2 : 1	0
2.5 : 1	1
3 : 1	2

For example, to change Barcode Thick to Thin Ratio to *2.5 : 1*, add the following line of code to modify the object labeled as *Barcode1*:

```
mh.SetStringProp "Barcode1", "BarcodeThickToThinRatio", "1"
```

AddCircle method

The AddCircle control adds a new circle object to the current drawing centered at (X1, Y1) with the specified radius using the current default object properties set in the WinMark Pro application. The string entered as *Object Name* is assigned as the object name for the new circle. The control method is formatted as:

AddCircle < ObjectName as String, X1 as Float, Y1 as Float, radius as Float > BOOLEAN

Visual Basic example:

```
' create a new circle centered at (0, 0) with a radius of
1 inch (2 inch diameter)

mh.AddCircle "CircleObject", 0, 0, 1

mh.Redraw
```

AddImport method

The AddImport control adds a new imported graphic to the drawing using the current default object properties specified in the WinMark Pro application. If you are importing vector-style graphics files using WinMark Pro v2.× or 3.×, then you must have previously installed the *File Import Filters* library. The string entered as *Object Name* is assigned as the object name for the new imported graphic. The control method is formatted as:

AddImport < ObjectName as String, FileName as String> BOOLEAN

Visual Basic example:

```
mh.AddImport "NewPicture", "c:\pic.bmp"

mh.Redraw
```

Note: Before importing a vector graphic using ActiveX, you must open WinMark Pro and set the appropriate Vector Import Scale.

Note: When importing bitmap images using the AddImport method, images are always imported using WinMark's *Bayer Dithering* Color Reduction Method property. See the Tip below.

Tip: When dealing with vector and bitmap imports in an ActiveX interface, the best method is to import and set graphic properties using WinMark Pro's *Drawing Editor*. Save the mark file and then use you customized application or user-interface to manipulate mark parameters.

AddLine method

The AddLine control adds a new line object to the current drawing positioned so that (X1, Y1) is the top-left corner and (X2, Y2) is the bottom-right corner. The string entered as *Object Name* is assigned as the object name for the new line that is created using the current default object properties specified in the WinMark Pro application. The control method is formatted as:

AddLine < ObjectName as String, X1 as Float, Y1 as Float, X2 as Float, Y2 as Float > BOOLEAN

Visual Basic example:

```
' add new line from (-1, -1) to (1, 1)

mh.AddLine "LineObject", -1, -1, 1, 1

mh.Redraw
```

AddLineEx method

The AddLineEx control adds a new line object to the current drawing positioned so that (X1, Y1) is the top-left corner and (X2, Y2) is the bottom-right corner. Power, Velocity, and the number of Mark Passes are also defined for the object. The string entered as *Object Name* is assigned as the object name for the new line that is created using the current default object properties specified in the WinMark Pro application. Remember to set Power, Velocity and Mark Passes values, otherwise the new polyline object is marked using default object property values.

The control method is formatted as:

AddLineEx < ObjectName As String, X1 As Single, Y1 As Single, X2 As Single, Y2 As Single, Power as Single, Velocity As Single, nPasses As Long > INTEGER

Visual Basic example:

```
' add new line from (-1, -1) to (1, 1) and mark at 80% power,
25 inches per second, using 1 pass

mh.AddLineEX "LineExObject", -1, -1, 1, 1, 80, 25, 1

mh.Redraw
```

AddPolyLine method

The AddPolyLine control adds a new polyline object to the current drawing using the current default object properties specified in the WinMark Pro application. The string entered as *Object Name* is assigned as the object name for the new polyline object.

nPts is the number of X-Y coordinate pairs and *x* is the floating point starting address of the X-Y array. If *bClosed* is True, then a line is added from the endpoint coordinate to the starting point coordinate to close the polyline. When *bClosed* is False, the polyline ends at the endpoint coordinates.

The control method is formatted as:

AddPolyLine < ObjectName As String, nPts As Integer, x As Single, bClosed As Boolean > INTEGER

Visual Basic example:

```
Private Type PLARRAY
  X As Single
  Y As Single
End Type

Dim Polyline() As PLARRAY
Dim T As Single ‘angle in degrees
Dim E As Single ‘eccentricity of ellipse 0 -> 1
Dim N As Integer ‘number of co-ordinate pairs in array
Dim I As Integer

‘ Draw an ellipse
‘ formula: for t = 0 to 360 degrees, 0 < e < 1,
‘ x = COS(t), y = ((1 - e^2)^.5)SIN(t)

N = 72
E = 0.5
ReDim Polyline(N)
For I = 0 To N
  T = (I * 2 / N) * (3.141592654) ‘ convert degrees to radians
  Polyline(I).X = txtScale.Text * Cos(T)
  Polyline(I).Y = txtScale.Text * (((1 - E ^ 2) ^ 0.5) * Sin(T))
Next I
mh.AddPolyLine “Polyline1”, N + 1, Polyline(0).X, False
mh.Redraw
```

Visual C++ example:

```
' First dimension an array of nPts containing X and Y coordinates
' for each point of the polyline.

  Dim A [2x nPts]
  A[0] = 1st X coordinate point
  A[1] = 1st Y coordinate point

struct FPT {

  float x;

  float y;

} PolyLine[72];

float fRadius = 1.0f, PI = 3.141592 ;

/* Approximate a circle with 5-degree line segments */

for( int ang=0 ; ang < 360 ; ang+= 5 )

{

 PolyLine[ ang ].x = fRadius * (float)cos( (double)ang*PI/180 ) ;

 PolyLine[ ang ].y = fRadius * (float)sin( (double)ang*PI/180 ) ;

}

/* Get the Client Rectangle */

CRect ClientRect;

GetDlgItem(IDC_SHOWMARK)->GetWindowRect(&ClientRect);

ScreenToClient(&ClientRect);

/* Create the ActiveX control: See CWnd::CreateControl in MSDN
help for a discussion on Create */

CSynMhAtx m_winMarkCtl;

BOOL bRet = m_winMarkCtl.Create(NULL, WS_VISIBLE , ClientRect,
this, IDC_SYNMHATX);

/* Load a drawing file */

m_winMarkCtl->LoadDrawing( "Default.mkh" );

/* Add the PolyLine */

m_winMarkCtl->AddPolyLine( "NewPoly", sizeof(PolyLine)/
sizeof(PolyLine[0]), &PolyLine[0].x, TRUE );
```

AddPolyLineEx method

The AddPolyLineEx control adds a new polyline object to the current drawing using the current default object properties specified in the WinMark Pro application. Power, Velocity, and the number of Mark Passes are also defined for the object. The string entered as *Object Name* is assigned as the object name for the new polyline object.

nPts is the number of X-Y coordinate pairs and *x* is the floating point starting address of the X-Y array. If *bClosed* is True, then a line is added from the endpoint coordinate to the starting point coordinate to close the polyline. When *bClosed* is False, the polyline ends at the endpoint coordinates.

The control method is formatted as:

AddPolyLineEx < ObjectName As String, nPts As Integer, x As Single, bClosed As Boolean, power As Single, velocity As Single, nPasses As Long > INTEGER

Visual Basic example:

See sample VB code for the AddPolyLine method above. Remember to set Power, Velocity and Mark Passes values after defining a *bClosed* value in the argument, otherwise the new polyline object is marked using default object property values.

Visual C++ example:

See sample VC++ code for the AddPolyLine method above. Remember to set Power, Velocity and Mark Passes values after defining a *bClosed* value in the argument, otherwise the new polyline object is marked using default object property values.

AddPreviewWnd method

The AddPreviewWnd control opens the WinMark Pro ActiveX form showing the *Drawing Canvas*. The control method is formatted as:

AddLine < pParent as Long, resourceID as Long > VOID

Visual C++ example:

```
m_winMarkCtrl.AddPreviewWnd((long*) this, IDC_SHOWMARK);
```

AddRect method

The AddRect control adds a new rectangle object to the current drawing using (X1, Y1) as the top-left corner and (X2, Y2) as the bottom-right corner using the current default object properties set in the WinMark Pro application. The string entered as *Object Name* is assigned as the object name for the new rectangle. The control method is formatted as:

AddRect < ObjectName as String, X1 as Float, Y1 as Float, X2 as Float, Y2 as Float > BOOLEAN

Visual Basic example:

```
' add a new rectangle from (-1, -1) to (1, 1)

mh.AddRect "RectangleObject", -1, -1, 1, 1

mh.Redraw
```

AddSpot method

The AddSpot control adds a new laser spot object to the current drawing centered at (X, Y) using the current default object properties set in the WinMark Pro application. The string entered as *Object Name* is assigned as the object name for the new rectangle. The control method is formatted as:

AddSpot < ObjectName as String, X as Float, Y as Float > BOOLEAN

Visual Basic example:

```
' add a new spot at (0, 0)

mh.AddSpot "SpotObject", 0, 0

mh.Redraw
```

AddText method

The AddText control adds a new text object to the current drawing at (X, Y) using the current default object properties set in the WinMark Pro application. The string entered as *Object Name* is assigned as the text object name. *Text Caption* is assigned as the text caption for the new object. The control method is formatted as:

AddText < ObjectName as String, X as Float, Y as Float, TextCaption as String > BOOLEAN

Visual Basic example:

```
' add a new text object at (0, 0)

mh.AddText "TextObject1", 0, 0, "Power by SYNRAD"

mh.Redraw
```

Tip: To specify a Font Type, either *Stroke* or *True Type* text, refer to the following chart:

Font Type	Property value
True Type	0
Stroke	1

For example, to specify True Type for the object labeled *Text1*, add the following line of code:

```
mh.SetStringProp "Text1", "FontType", "0"
```

Tip: To specify Text Justification for either *Stroke* or *True Type* text, refer to the following chart:

Text Justification	Property value
Left	0
Center	1
Right	2

For example, to specify center justification for an object labeled *Text1*, add the following code:

```
mh.SetStringProp "Text1", "TextJustification", "1"
```

Tip: To change the Auto Text Type of a text object, refer to the following chart:

Auto Text Type	Property value
None	0
Serial Number Text	1
Date Code Text	2
Disk File Text (sequential)	3
Serial Port Text	4
User Entry Text	5
Disk File Text (fixed)	6
From Another Object	7

For example, to define an automated Text Caption for the object labeled *Text1* using data sent to the computer's serial port, add the following line of code:

```
mh.SetStringProp "Text1", "AutoTextType", "4"
```

Tip: To specify when an auto text update should occur, refer to the following chart:

Execute Auto Text When	Property value
Before Mark	0
After Mark	1

For example, to specify auto updating after each mark, add the following line of code:

```
mh.SetStringProp "Text1", "ExecuteAutoTextWhen", "1"
```

Tip: To anchor text to a specific position with respect to the bounding box, Fixed To Position, refer to the following chart:

Fixed To Position	Property value
Top Left	0
Top Middle	1
Top Right	2
Right Middle	3
Bottom Right	4
Bottom Middle	5
Bottom Left	6
Left Middle	7
Center	8
Radial Center*	9

* For radial text objects – not yet implemented.

For example, to anchor the object labeled *Text1* to the left middle of the bounding box, add the following line of code:

```
mh.SetStringProp "Text1", "FixedToPosition", "7"
```

AutoTestMark method

The AutoTestMark control downloads the current .mkh mark file to the marking head's *Custom Test Mark* memory (see the *Fenix Laser Marker Operator's Manual* or the *FH Series Marking Head Operator's Manual* for details about the Custom Test Mark feature).

Set *bNoAuto* to True when a Manual test mark is desired. When *bNoAuto* is False, an Auto test mark is specified. Index test marks may be Manual or Auto; tracking test marks must be defined as Auto.

Set *bIndexMark* to True for an Auto Index test mark. Set *bIndexMark* False to specify an Auto Tracking test mark (only functional when marking head is a Tracker model).

Set *nDownload* to True to download the current mark file and current test mark settings; set to False to change settings without downloading a new mark file.

The control method is formatted as:

AutoTestMark < bNoAuto as Boolean, bIndexMark as Boolean, nDownload as Boolean > BOOLEAN

Visual Basic example:

The following code is used on a form (Figure 20-1) that has a Download File checkbox (chkDownload), and two groups of option buttons: Manual (optMan) or Auto (optAuto), and Index (optIndex) or Tracking (optTracker). Because of the grouping, if optAuto is selected (True), optMan is then false, and the same is true for Index and Tracking.

```
' first set the mark file to Index or Tracking
mh.SetBoolProp "Drawing", "TrackMarkingObject", optTracker
' now download the test mark data
mh.AutoTestMark optMan, optIndex, chkDownload
```

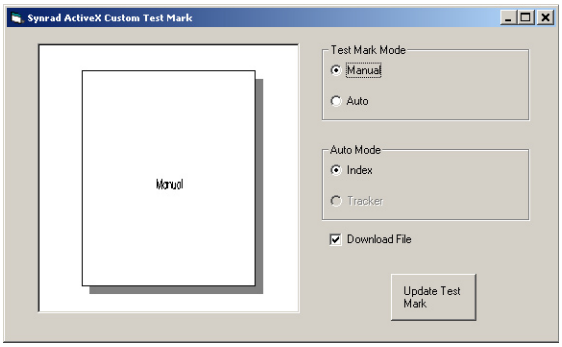


Figure 20-1 Sample AutoTestMark form

CenterSelection method

The CenterSelection control centers the current selection set in the mark field. The control method is formatted as:

CenterSelection < none > BOOLEAN

ClearSelection method

The ClearSelection control deselects all objects contained in the current selection set. The control method is formatted as:

ClearSelection < none > BOOLEAN

Visual Basic example:

```
' select and rotate the Text1 object
mh.SelectObject "Text1"
mh.RotateSelection 15
```

```
mh.ClearSelection
' Select the Barcode1 object and move to X2, Y-3
mh.SelectObject "Barcode1"
mh.MoveSelectionTo 2,-3
mh.ClearSelection
mh.Redraw
```

ConfigureAbort method

The ConfigureAbort control allows you to configure the abort sequence. You can enable or disable the ability to abort using the *bEnabled* parameter. SYNRAD strongly recommends that you enable this parameter by setting it to *1* (TRUE). By default, the ESC key aborts marking, however *iKey* can be set to any virtual key code. When the user aborts marking, a dialog box asks for confirmation. Use the *bShow-Dialog* parameter to turn this feature on or off.

⚠ Warning

serious personal injury

When marking with ActiveMark (ActiveX/OLE) control, the Windows priority assigned to WinMark Pro and to the MarkDrawing method is high enough that only the Escape (ESC) key is recognized by the Windows operating system. Any other key (except ESC) assigned as the abort key by the ConfigureAbort method is ***NOT*** recognized during a mark session.

You should assign only the ESC key as the abort method to halt lasing during a mark session.

The control method is formatted as:

ConfigureAbort < bEnabled as Boolean, iKey as Integer, bShowDialog as Boolean > VOID

Visual Basic example:

```
mh.ConfigureAbort 1, VK_ESCAPE, 1
```

DeleteObject method

The DeleteObject control deletes the specified object, referenced by *Object Name*, from the current drawing. The control method is formatted as:

DeleteObject < ObjectName as String > BOOLEAN

Visual Basic example:

```
' delete the text object named "Text1"

mh.DeleteObject "Text1"

mh.Redraw
```

DeleteSelection method

The DeleteSelection control deletes all objects in the current selection set from the active drawing. The control method is formatted as:

DeleteSelection < none > BOOLEAN

Visual Basic example:

```
' delete all objects in the current drawing

mh.SelectAll

mh.DeleteSelection

mh.Redraw
```

EraseDrawing method

The EraseDrawing control removes all objects in the current drawing. The control is formatted as:

EraseDrawing < none > BOOLEAN

Note: In WinMark Pro versions 4.1.1.4402 and above, the EraseDrawing method removes all entities from the drawing while leaving *Drawing* properties unchanged.

In versions prior to build 4402, the EraseDrawing control opened a new blank form after erasing the current drawing. This new form however, used Field Height and Field Width dimensions from a DH lens (square mark field) even when the previous drawing was based on an FH lens (rectangular mark field). When using versions older than 4402 we recommend using SelectAll and DeleteSelection methods to delete all objects in the active drawing.

Visual Basic example:

```
mh.AddRect "rect1", 0, 0, 1, 1

mh.AddRect "rect2", -1, -1, 0, 0

mh.EraseDrawing

mh.Redraw
```

GetBoolProp method

The GetBoolProp control gets a Boolean property from the current drawing. The control method is formatted as:

GetBoolProp < ObjectName as String, PropertyName as String > BOOLEAN

Visual Basic example:

```
Dim flag as Boolean

flag = mh.GetBoolProp "Text1", "MarkObject"
```

GetDigitalBit method

The GetDigitalBit control reads the state of the specified digital input connected to an FH Series marking head, Fenix Laser Marker, or Fiber Link Controller Card (FLCC). See Chapter 12, *Automation basics*, for a chart showing valid FLCC or marking head input bits. Use these digital input signals to handshake with external hardware, start marking, or perform other input functions. The control method is formatted as:

GetDigitalBit < nBit as Integer > BOOLEAN

Important Note: When developing your automated control system using ActiveX control methods, do not place input automation into the mark file. Instead, perform any input automation within your ActiveX control program. By doing this, your control program remains in charge of pre- mark and post-mark automation and WinMark's MarkDrawing control method only maintains control of the computer's operating system long enough to control laser on/off commands during the actual marking process. Performing input automation within the mark file may prevent the ActiveX control program from regaining control of the operating system if the specified input state fails to occur.

Visual Basic example 1:

```
' get state of Bit 0 (IN0)

mh.GetDigitalBit 0

' get state of Bit 1 (IN1)

mh.GetDigitalBit 1
```

Visual Basic example 2:

Note: This is an ActiveX version of the basic automation sequence described in Chapter 12, *Automation basics*, using GetDigitalBit and SetDigitalBit control methods.

```
' Event 0 - set OUT4 - laser is off; system ready for new part
mh.SetDigitalBit 4, True
' Event 1 - wait for "start mark" signal on IN0
lblStatus.Caption = "Waiting for Bit 0"
lblStatus.Refresh
Do While Not (mh.GetDigitalBit(0))
  DoEvents
Loop
' Event 2 - clear OUT4; update Status label - laser is on
mh.SetDigitalBit 4, False
lblStatus.Caption = "Marking"
lblStatus.Refresh
' update mark file objects as required
' update the display and mark
mh.Redraw
mh.MarkDrawing
' Event 3 - verify IN0 is clear to avoid another mark on part
lblStatus.Caption = "Waiting for Bit 0"
lblStatus.Refresh
Do While (mh.GetDigitalBit(0))
  DoEvents
Loop
' Event 4 - mark done; set OUT4 - laser off; ready for new part
mh.SetDigitalBit 4, True
' update captions and go back to ready state
lblStatus.Caption = "Ready"
cmdMark.Caption = "Mark"
' add loop code as needed to mark more than one part
```

GetFloatProp method

The GetFloatProp control gets a float property value from the current drawing. The control method is formatted as:

GetFloatProp < ObjectName as String, PropertyName as String > FLOAT

Visual Basic example:

```
Dim value as single

value = mh.GetFloatProp "Text1", "ObjectLeftPos"
```

GetHeadInformation method

The GetHeadInformation control returns Card type, Head type, Data type, Firmware version, and PCI version data as integers. Integer results correspond to the following values:

Card type:	1	No Card
	2	DDA06 (DAC) Card
	3	ISA-bus Fiber Link Controller Card (FLCC), talk only
	4	ISA-bus FLCC, talk/listen
	5	PCI-bus FLCC
Head type:	0	No Data
	1	SH or unknown
	2	DH
	3	Fenix Index
	4	FH Index
	5	Fenix Tracker
	6	FH Tracker
Data type:	0	No Data
	1	12 Bit data
	2	16 Bit data
Head firmware version:	0	No Data
	×	(firmware version 3–20)
PCI card revision ID:	×	(revision ID)

The control method is formatted as:

GetHeadInformation < Card ByRef Integer, Head ByRef Integer, Data ByRef Integer, Firmware ByRef Integer, PCIrev ByRef Integer > STRING

Note: Head, Data, and Firmware values will return as zero (0) if the marking head is not powered up or not connected to the FLCC.

Visual Basic example:

```
Dim Card as integer ;FLCC card type code listed above

Dim Head as integer ;marking head type code listed above

Dim Data as integer ;data type used by head listed above

Dim Firmware as integer ;marking head firmware version

Dim PCIrev as integer ;PCI-bus FLCC firmware version ID

Dim LensType as String ;lens type string (e.g. "FH/Fenix 80 mm"
or "SH/DH 200 mm" as stored in the computer's System Registry.

LensType = mh.GetHeadInformation Card, Head, Data, Firmware,
PCIrev
```

GetIntProp method

The GetIntProp control gets an integer property value from the current drawing. The control method is formatted as:

GetIntProp < ObjectName as String, PropertyName as String > INTEGER

Visual Basic example:

```
Dim value as Integer

value = mh.GetIntProp "Text1", "Resolution"
```

GetMarkErrorCode method

The GetMarkErrorCode control returns the last error code, if any, generated during marking. If multiple errors occur during a single mark session, only the last non-zero error is returned.

Integer error code results are:

0	No Mark Error	– mark successfully completed.
1	No Response From Head	– no response through fiber link connection.
2	Line Speed Too Fast To Finish	– target mark area has moved outside the marking window (Tracking mode only).
3	Fiber Link Overrun Error	– FH or Fenix marking head could not keep up with the incoming data stream.
4	Line Speed Too Fast; Missed Start	– part sensor input received before previous mark completed (Tracking mode only).
5	Encoder Continuity Error	– encoder pulses at FH head corrupted (Tracking mode only).
6	Marking Head Not Ready	
7	Bad End-Of-Mark Response	– FH or Fenix marking head did not respond correctly at end of mark.
8	Marking Head Not Powered Up	– FH or Fenix head not responding to FLCC query; head is not powered up or fiber link disconnected.
9	Invalid Drawing	– .mkh mark file corrupted.
10	TestMark Mode Fiber Link Error	– head does not respond on entry to AutoTestMark download method. Check that head is powered up, fiber link cable connected, etc.
11	TestMark Mode Clear Error	– cannot clear the current test mark mode.
12	TestMark Mode Set Error	– cannot set the requested test mark mode.
13	TestMark Transmit Error	– error during transmission of test mark to head.
14	TestMark Too Big	– test mark is too large to fit in FH or Fenix memory.
15	Bitmap Too Fast To Mark Error	– the product of mark Velocity times the Resolution is too large for the *Continuous* marking method. Velocity has been reduced to an acceptable value.
16	Marking Head Not Responding	– FH or Fenix head not responding to commands at start of mark; possibly not powered up.

The control method is formatted as:

GetMarkErrorCode < none > INTEGER

Visual Basic example:

```
Dim E As Integer
Dim Err As String
mh.MarkDrawing
E = mh.GetMarkErrorCode
If Not E = 0 Then
  Select Case E
    Case 1
      Err = "No Response From Head"
    Case 2
      Err = "Line Speed Too Fast To Finish"
    Case 3
      Err = "Fiber Link Overrun"
    Case 4
      Err = "Line Speed Too Fast - Missed Start"
    Case 5
      Err = "Encoder Continuity Error"
    Case 6
      Err = "Marking Head Not Ready"
    Case 7
      Err = "Bad End Of Mark Response"
    Case 8
      Err = "Marking Head Not Powered Up"
    Case 9
      Err = "Invalid Drawing"
    Case Else
      Err = Str(E)
  End Select
  MsgBox "The following error was detected:" & vbCr & vbCr & Err
End If
```

GetObjectBounds method

The GetObjectBounds control returns the bounding rectangle for the specified object. The control method is formatted as:

GetObjectBounds < ObjectIndex as Integer, Left ByRef Float, Top ByRef Float, Width ByRef Float, Height ByRef Float > BOOLEAN

Visual Basic example:

```
Dim x as single
Dim y as single
Dim w as single
Dim h as single
mh.GetObjectBounds 0, x, y, w, h
```

GetObjectCount method

The GetObjectCount control returns the object count for the currently active drawing. The control method is formatted as:

GetObjectCount < none > SHORT

Visual Basic example:

```
Dim n as Integer
n = mh.GetObjectCount
```

GetObjectIndex method

The GetObjectIndex control returns the index number of an object in the drawing. The control method is formatted as:

GetObjectIndex < ObjectName as String > INTEGER

Visual Basic example:

```
mh.EraseDrawing
mh.AddRect "rect", 0, 0, 1, 1
Dim nObject as Integer
nObject = mh.GetObjectIndex "rect"
```

GetObjectName method

The GetObjectName control returns the name of an object by index number. Note that the *Drawing* object is not accessible by the GetObjectName method. The control method is formatted as:

GetObjectName < ObjectIndex as Integer > STRING

Visual Basic Example

Note: The following code is used on a form that has a list box (*lstObjects*) and a command button (*cmdListObjects*) which allows the user to list the objects currently residing on the loaded drawing. *Label1* displays the number of objects found.

```
Private Sub cmdListObjects_Click()
Dim I As Integer, N As Integer
Dim objects() As String
Dim Obj As String
lstObjects.Clear
N = mh.GetObjectCount
ReDim objects(N) As String
For I = 0 To N - 1
  Obj = mh.GetObjectName(I)
  lstObjects.AddItem Obj, I
Next I
Label1.Caption = N & " items"
End Sub
```

GetObjectPos method

The GetObjectPos control returns an XY reference position for an object in the drawing. The reference position is the point around which the selected object or objects is rotated or transformed. Integer results correspond to the following positions:

0 Top Left

1 Top Right

2 Bottom Right

3 Bottom Left

4 Center

The control method is formatted as:

GetObjectPos < ObjectIndex as Integer, Position as Int, x ByRef Float, y ByRef Float> BOOLEAN

Visual Basic example:

```
Dim x as single
Dim y as single
' get top left position coordinates
mh.GetObjectPos 0, 0, x, y
' get center position coordinates
mh.GetObjectPos 0, 4, x, y
```

GetObjectType method

The GetObjectType control returns the specific object type of a particular object. Integer results correspond to the following object types:

0	Unknown
1	Group
2	Text
3	Polyline
4	Raster Bitmap
5	Vector Import
6	Arc
7	Bar Code
8	Spot
9	2D Bar Code

The control method is formatted as:

GetObjectType < ObjectIndex as Integer > SHORT

GetSelectionBounds method

The GetSelectionBounds control returns the bounding rectangle dimensions for the current selection set. The control method is formatted as:

GetSelectionBounds < Left ByRef Float, Top ByRef Float, Width ByRef Float, Height ByRef Float > BOOLEAN

Visual Basic example:

```
Dim x as single
Dim y as single
Dim w as single
Dim h as single
mh.SelectObject "rect"
mh.GetSelectionBounds x, y, w, h
```

GetStringProp method

The GetStringProp control gets a string property value from the current drawing. The control method is formatted as:

GetStringProp < ObjectName as String, PropertyName as String > STRING

Visual Basic example:

```
Dim value as String
value = mh.GetStringProp "Text1", "TextCaption"
```

LoadDrawing method

The LoadDrawing control loads the specified WinMark Pro .mkh mark file into the control. The drawing is not displayed until the Redraw method is called. If the ProgressCtrlHandle is set to display a progress bar, then the progress is indicated as the file load proceeds. The control method is formatted as:

LoadDrawing < Filename as String > BOOLEAN

Visual Basic example:

```
mh.DrawingIndex = 0
mh.ProgressCtrlHandle = ctlProgressBar.hWnd
mh.LoadDrawing "FactoryTestMark1.mkh"
mh.Redraw
```

MarkDrawing method

The MarkDrawing control marks the current drawing. If the ProgressCtrlHandle is set to display a progress bar, then progress is indicated as the mark proceeds.

A True returned value indicates successful completion of the mark. A False returned value indicates that either an error occurred during marking or that the mark was aborted. See GetMarkErrorCode to determine which error occurred.

Important Note: When developing an automated control system using ActiveX control methods, do not place input automation into the mark file. Instead, perform any input automation functions within the ActiveX control program. By doing this, your control program remains in charge of pre- mark and post-mark automation and WinMark's MarkDrawing control method only maintains control of the operating system long enough to control laser on/off commands during the actual marking process. Performing input automation within the mark file may prevent the ActiveX control program from regaining control of the operating system if the specified input state does not occur.

The control method is formatted as:

MarkDrawing < none > BOOLEAN

Visual Basic example:

```
' select the active drawing
mh.DrawingIndex = 0
' enable the progress bar
mh.ProgressCtrlHandle = ctlProgressBar.hWnd
' load the drawing
mh.LoadDrawing "FactoryTestMark1.mkh"
' refresh the screen display
mh.Redraw
' mark the drawing
mh.MarkDrawing
```

MirrorSelection method

The MirrorSelection control mirrors all objects in the current selection set along the X- or Y-axis. Objects are mirrored about the center point of the selection set. The control method is formatted as:

MirrorSelection < FlipX as Boolean, FlipY as Boolean > BOOLEAN

Visual Basic example:

```
' select all objects in drawing and mirror across X-axis
mh.SelectAll
mh.MirrorSelection True, False
mh.Redraw
```

MoveSelectionTo method

The MoveSelectionTo control moves all objects in the current selection set to the desired (X, Y) location. Specifically, the top left position of the selection set is moved to the indicated location. The control method is formatted as:

MoveSelectionTo < XPos as Float, YPos as Float > BOOLEAN

Visual Basic example:

```
' select all objects in the current drawing
mh.SelectAll
' move the selection set to (-1, -1)
mh.MoveSelectionTo -1, -1
mh.Redraw
```

Redraw method

The Redraw control redraws the display to reflect any changes made to the current drawing via ActiveX control. The control method is formatted as:

Redraw < none> VOID

Visual Basic example:

```
' select all objects in drawing, rotate 45 degrees, and redraw
mh.SelectAll
mh.RotateSelection 45
mh.Redraw
```

Reset method

The Reset control removes all entities in the current drawing and resets *Drawing* properties to the default values set on the *Tools/General Settings…/Application Settings* tab. The control is formatted as:

Reset < none > BOOLEAN

Important Note: Prior to WinMark Pro build 4.1.1.4402, the Reset control opened a new blank form after erasing the current drawing. This new form however, used Field Height and Field Width dimensions from a DH lens (square mark field) even when the previous drawing was based on an FH lens (rectangular mark field).

Visual Basic example:

```
mh.AddRect "rect1", 0, 0, 1, 1

mh.AddRect "rect2", -1, -1, 0, 0

mh.Reset

mh.Redraw
```

RotateSelection method

The RotateSelection control rotates all objects in the current selection set by the specified angle. Positive angles are rotated clockwise while negative angles are rotated counterclockwise. The center of rotation is the center point of the current selection set. The control method is formatted as:

RotateSelection < Angle as Integer > BOOLEAN

Note: Only integer rotation values are permitted using the RotateSelection method.

Important Note: Imported bitmap objects only rotate in 90 degree increments.

Tip: Although you can rotate bitmap objects only in 90 degree intervals, you can rotate the entire drawing canvas by any interval using the *Drawing* object's Field Rotation property. Remember that this rotation will affect all objects in the drawing.

Visual Basic example:

```
' select all objects in the drawing and rotate 45 degrees

mh.SelectAll

mh.RotateSelection 45

mh.Redraw
```

Visual C++ example:

```
' rotate the current selection set by 45 degrees

status = m_winMarkCtrl.RotateSelection(45);
```

SaveDrawing method

The SaveDrawing control saves the current drawing. If the ProgressCtrlHandle is set to display a progress bar, then the Save progress is indicated as the mark proceeds. The control method is formatted as:

SaveDrawing < Filename as String > BOOLEAN

Visual Basic example:

```
' select the active drawing

mh.DrawingIndex = 0

' enable the progress bar

mh.ProgressCtrlHandle = ctlProgressBar.hWnd

' load the drawing

mh.LoadDrawing "FactoryTestMark1.mkh"

' save as a new drawing

mh.SaveDrawing "NewTestMark.mkh"
```

ScaleSelection method

The ScaleSelection control scales the selection set in the current drawing. A scale value of 1.1 will scale the selection set up in size by 10%; a value of 0.9 will scale the selection set down by 10%.

Note: Scaling a selection non-proportionally results in objects being stretched or compressed in 1-axis.

The control method is formatted as:

ScaleSelection < XScale as Float, YScale as Float > BOOLEAN

Visual Basic example:

```
' select all objects in drawing and scale proportionally by 50%

mh.SelectAll

mh.ScaleSelection 0.5, 0.5

mh.Redraw
```

SelectAll method

The SelectAll control selects all objects in the current drawing into a selection set. Each drawing maintains a current selection set that is used to modify one or more objects. The control method is formatted as:

SelectAll < none > BOOLEAN

Visual Basic example:

```
' activate a specific drawing
mh.DrawingIndex = 0
' select all objects in the current drawing
mh.SelectAll
' rotate the current selection set
mh.RotateSelection 45
mh.Redraw
```

SelectObject method

The SelectObject control selects an object by its name in the current drawing. The control method is formatted as:

SelectObject < ObjectName as String> BOOLEAN

Visual Basic example:

```
' select the Text1 object into the selection set
mh.SelectObject "Text1"
' rotate the Text1 object 90 degrees
mh.RotateSelection 90
mh.Redraw
```

SelectObjectIndex method

The SelectObjectIndex control selects individual objects by their index number. The control method is formatted as:

SelectObjectIndex < ObjectIndex as Integer > BOOLEAN

Visual Basic example:

```
' delete existing objects
mh.SelectAll
mh.DeleteSelection
' add first object (object index 0)
mh.AddRect "Rectangle1", 0, 0, 1, 1.5
' add second object (object index 1)
mh.AddCircle "Circle1", 0, 0, 0.5
mh.SelectObjectIndex 0
' rotate Rectangle1 by 90 degrees
mh.RotateSelection 90
mh.Redraw
```

SetBoolProp method

The SetBoolProp control assigns a Boolean value (1 or 0) to a specific property of an object in the current drawing. *ObjectName* must reference a valid object in the current drawing and *PropName* must be an available property for that object. The control method is formatted as:

SetBoolProp < ObjectName as String, PropName as String, Value as Boolean > BOOLEAN

Visual Basic example:

```
' enable bolding of Text1 object
mh.SetBoolProp "Text1", "'TextBold", 1
' enable raster fill of Text1 object
mh.SetBoolProp "Text1", "TextFilled", 1
' disable italics of Text1 object
mh.SetBoolProp "Text1", "TextItalics", 0
mh.Redraw
```

SetDigitalBit method

The SetDigitalBit control sets a specific output bit on the digital output connected to an FH Series marking head, Fenix Laser Marker, or Fiber Link Controller Card (FLCC). See Chapter 12, *Automation basics*, for a chart showing valid FLCC or marking head output bits. Use these digital output signals to handshake with external hardware or parts handling equipment. The control method is formatted as:

SetDigitalBit < nBit as Integer, BOOL bValue > VOID

Visual Basic example 1:

```
' set Bit 4 (OUT4)

mh.SetDigitalBit 4, True

' clear Bit 5 (OUT5)

mh.SetDigitalBit 5, False
```

Visual Basic example 2:

Note: This is an ActiveX version of the basic automation sequence described in Chapter 12, *Automation basics*, using GetDigitalBit and SetDigitalBit control methods.

```
' Event 0 - set OUT4 to indicate laser is off and the
' marking system is ready for a new part

mh.SetDigitalBit 4, True

' Event 1 - wait for "start mark" command on IN0

lblStatus.Caption = "Waiting for Bit 0"

lblStatus.Refresh

Do While Not (mh.GetDigitalBit(0))

  DoEvents

Loop

' Event 2 - clear OUT4 & update Status label to indicate laser on

mh.SetDigitalBit 4, False

lblStatus.Caption = "Marking"

lblStatus.Refresh

' update mark file objects as required

' update the display and mark

mh.Redraw

mh.MarkDrawing
```

```
' Event 3 - verify that IN0 is clear to avoid looping more than
' once on one "start mark" command

lblStatus.Caption = "Waiting for Bit 0"

lblStatus.Refresh

Do While (mh.GetDigitalBit(0))

  DoEvents

Loop

' Event 4 - marking done - set OUT4 to indicate laser off;
' ready for new part

mh.SetDigitalBit 4, True

' update captions and go back to ready state

lblStatus.Caption = "Ready"

cmdMark.Caption = "Mark"

' add loop code as needed to mark more than once...
```

SetFloatProp method

The SetFloatProp control assigns a floating-point value to a specific property of an object in the current drawing. *ObjectName* must reference a valid object in the current drawing and *PropName* must be an available property for that object. The control method is formatted as:

SetFloatValue < ObjectName as String, PropName as String, Value as Float > BOOLEAN

Visual Basic example:

```
' move Text1 object to XY coordinate 1.5, 1.25

mh.SetFloatProp "Text1", "ObjectLeftPos", 1.5

mh.SetFloatProp "Text1", "ObjectTopPos", 1.25

' set Text1 font height to 0.25 inches

mh.SetFloatProp "Text1", "TextHeight", 0.25

mh.Redraw
```

SetIntProp method

The SetIntProp control assigns an integer value to a specific property of an object in the current drawing. *ObjectName* must reference a valid object in the current drawing and *PropName* must be an available property for that object. The control method is formatted as:

SetIntProp < ObjectName as String, PropName as String, Value as Integer > BOOLEAN

Visual Basic example:

```
' rotate marking output by 45 degrees
mh.SetIntProp "Drawing", "FieldRotation", 45
' set 750 DPI resolution for Text1 object
mh.SetIntProp "Text1", "Resolution", 750
' set 2 mark passes on Text1 object
mh.SetIntProp "Text1", "MarkPasses", 2
mh.Redraw
```

SetOffset method

The SetOffset control offsets the position of the *Drawing Canvas* within the preview window. The offset does not affect object position when marking. The control method is formatted as:

SetOffset < x As Single, y As Single > VOID

Visual Basic example:

```
' offset Drawing Canvas position by 0.75 inches in X direction
mh.SetOffset 0.75, 0
```

SetStringProp method

The SetStringProp control assigns a string value to a specific property of an object in the current drawing. *ObjectName* must reference a valid object in the current drawing and *PropName* must be an available property for that object. The control method is formatted as:

SetStringProp < ObjectName as String, PropName as String, Value as String> BOOLEAN

Visual Basic example:

```
' set the text caption for the Text1 object
mh.SetStringProp "Text1", "TextCaption", "Power By SYNRAD!"
' select a true type font for the Text1 object
```

```
mh.SetStringProp "Text1", "TrueTypeFont", "Times New Roman"
mh.Redraw
```

Tip: The SetStringProp method can assign values to any valid property including those that require floating point, integer, or Boolean values. The string value will be converted to the appropriate type in these cases.

In cases where your ActiveX mark file contains non-Event Builder output automation, you can use the SetStringProp to modify the expected output states for the *Drawing* object's Set Digital Before Piece, Set Digital Before Mark, and Set Digital After Mark automation properties. SetStringProp can control input or output bits on any of the digital I/O lines connected to an FH Series marking head, Fenix Laser Marker, or Fiber Link Controller Card (FLCC).

Note: We do not recommend placing input automation into the mark file. Instead, perform any input automation functions within the ActiveX control program using the GetDigitalBit control method. By doing this, your control program remains in charge of pre-mark and post-mark automation and WinMark's MarkDrawing method only maintains control of the operating system long enough to control laser on/off commands during the actual marking process. Performing input automation within the mark file may prevent the ActiveX control program from regaining control of the operating system if the specified input state does not occur.

Tip: I/O string characters used to force a specific input or output bit state are listed in the following chart:

I/O Bit State	String Character
Ignore	–
Clear	0
Set	1

Valid input bits are 0, 1, 2, 3, 4, 5, 6, and 7 although not all inputs may be supported by your marking hardware. Valid output bits are 1, 3, 4, 5, 6, and 7 although not all outputs may be supported by your marking hardware. See Chapter 12, *Automation basics*, for a chart showing valid input and output bits for your specific I/O interface.

To set *Output Bit 4* (*OUT4*) in the *Drawing* object's Set Digital After Mark automation property and *Ignore* (or leave unchanged) all other output bit states, add the following line of code:

```
mh.SetStringProp "Drawing", "SetDigitalAfterMark", "-","-","1","-
","-","-"
```

SetZoomFactor method

The SetZoomFactor control sets the magnification of the *Drawing Canvas* within the preview window. A zoom value of 2 zooms in by a factor of 200% of the normal view; a value of 0.5 zooms out to 50% of the normal view. The control method is formatted as:

SetZoomFactor < fFactor as Float > VOID

Visual Basic example:

```
' zoom to normal view
Private Sub Zoom_Normal
mh.SetZoomFactor 1
mh.Redraw
End Sub

' zoom in 200%
Private Sub Zoom_In
mh.SetZoomFactor 2
mh.Redraw
End Sub

' zoom out to 50%
Private Sub Zoom_Out
mh.SetZoomFactor 0.5
mh.Redraw
End Sub
```

TranslateSelection method

The TranslateSelection control re-positions objects in the current selection set by specifying X and Y offset values. While it is possible to move objects entirely off the *Drawing Canvas*, remember that the mark is not completed correctly when an object is placed partially on or fully off the *Drawing Canvas* and the MarkObject property is *Yes*. The control method is formatted as:

TranslateSelection < xOffset as Float, yOffset as Float > BOOLEAN

Visual Basic example:

```
' select all objects in drawing and offset to upper-left quadrant

mh.SelectAll

mh.TranslateSelection -1, -1

mh.Redraw
```

WritePropListToTextFile method

The WritePropListToTextFile control writes all properties of the object indicated by *ObjectName* into the file specified by *FileName*. The properties of the *Drawing* object can be written using "Drawing" for the *ObjectName* parameter. If the file already exists, data is appended to the end of the file; the file is created if it does not exist. The function returns the number of properties that were written to the file. The control method is formatted as:

WritePropListToTextFile < FileName as String, ObjectName as String > LONG

Visual Basic example:

```
Dim Count as Integer

Dim I as Integer

' call GetObjectCount to count the objects in a drawing

  Count = mh.GetObjectCount

' the Drawing object is not accessible using GetObjectName so...

  mh.WritePropListToTextFile "ActiveXProps.txt", "Drawing"

' now step through and write properties for all other objects

  If Not Count = 0 Then

    For I = 0 to (Count - 1)

      mh.WritePropListToTextFile "ActiveXProps.txt", (mh.
      GetObjectName(I))

    Next I

  End If
```

Tip: Customers using an Application Language other than English can print this list in order to compare English property names with corresponding property names translated according to the language specified by Application Language. Units of measure are printed using the currently selected units—inches, centimeters, or millimeters.

21 Troubleshooting

The *Troubleshooting* chapter includes subsections:

- Fiber Link Controller Card (FLCC)
- FH Series marking head
- Automation
- Graphics
- ActiveMark
- Laser Marking FAQ
- Resources

Fiber Link Controller Card (FLCC)

Symptom:

- When opened, WinMark Pro® returns a message indicating, "There is no interface card installed for this system."

Possible Causes:

- There is no Fiber Link Controller Card (FLCC) installed.

When using WinMark Pro in "demo" mode (when there actually is no FLCC installed), this message is normal. Click *OK* to continue with offline file development.

- The computer was not rebooted after installing WinMark Pro v4.0.

After installation, reboot the computer in order to load the new PCI Device Drivers. After rebooting, check that the FLCC driver is properly installed. Open the Windows® *Device Manager* and click the *View connections by type* radio button. Under *Laser Marking Devices*, look for an entry titled "FH/Fenix Laser Marking Devices".

- Both ISA-bus and PCI-bus FLCCs are installed.

Select the PCI card manually. Open WinMark Pro and go to the *Tools\General Settings…\Application Settings* tab. Locate the I/O Card Selection property (see Figure 21-1) and in the drop-down list, select "PCI Card ID# ×".

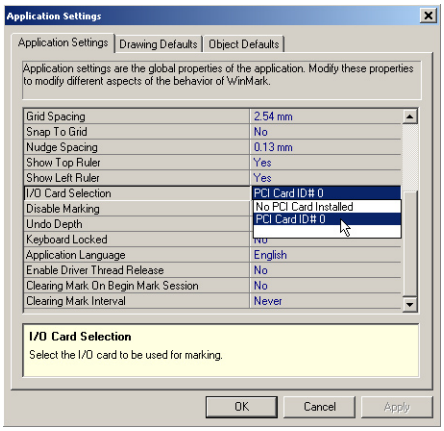


Figure 21-1 Application Settings tab

- The installed FLCC is an ISA-bus card.

WinMark Pro version 4 does not support ISA-bus FLCCs. You must install a PCI-bus FLCC or you may choose to install a version 2.× build of WinMark Pro instead.

WinMark Pro version 2 supports ISA-bus FLCCs running on Windows 98 or Windows 95. Because v2 is a legacy version, it does not contain the latest features found in v4 builds. Check the WinMark Pro CD or the WinMark Pro web site at http://www.winmark.com\download.htm for the latest released builds of v4 or v2.

Symptom:

- WinMark Pro or WinMark Launcher applications indicate that marking was successful, but the laser does not fire. The laser does not fire when the marking head's *Test Mark* pushbutton is pressed.

Possible Causes:

- The laser is not enabled.

Check that the laser's *Ready* light is illuminated. On Series 48 and Evolution lasers, check that the *Lase* indicator is glowing dimly (tickle mode) and that it appears brighter (lasing mode) as the mark is made. On Firestar lasers, check that *RDY* (Ready) and *SHT* (Shutter) indicators are lit and that the *LASE* indicator illuminates as the mark is being made. Refer to your laser's *Operator's Manual* if indicators do not illuminate as described.

Verify that the laser's shutter (if equipped) is open.

Ensure that all required safety interlocks and signals are present.

- The Fast Acting Safety Interlock (FASI) feature on the Fiber Link Card is enabled.

If the FASI feature is enabled (DIP switch #5 on the PCI-bus Fiber Link Controller Card set to "On"), then a 15–40 V signal must be applied to *Input Bit 3* (*IN3*) on the FH/Fenix™ marking head before the laser will fire.

Note: On older ISA-bus Fiber Link Controller Cards, the FASI feature is controlled by DIP switch # 6. Refer to the *Operator's Manual* that came with your particular marking head for details.

Symptom:

- A mark is made when the marking head's *Test Mark* pushbutton is pressed, but marking does not occur when commanded by WinMark Pro or WinMark Launcher.

Possible Causes:

- The fiber optic cable between the marking head and the FLCC is improperly connected or is damaged.

Refer to your marking head's *Operator's Manual* for information on verifying the fiber optic connection.

- The FLCC is not seated properly or the computer's PCI card slot is damaged.

Shut down the computer and reseat the FLCC in its PCI slot. If this fails to work, remove the Fiber Link Controller Card from its current card slot, and then reinstall the FLCC into a different PCI card slot.

- The FLCC has failed or is damaged.

If you determine that the FLCC is not functioning correctly, contact SYNRAD, Inc. or a SYNRAD Authorized Distributor for assistance.

FH Series marking head

Symptom:

- The *DC Power* cable is connected to the marking head, but the *ON* indicator is not illuminated.

Possible Causes:

- The proper DC voltage is not available from the DC power supply.

Ensure that +30 VDC at 3 amperes is available under full-load conditions.

- The fuse has blown.

Turn the fuse holder cover counterclockwise and remove the fuse. Inspect or test the fuse, replacing as necessary with the correct 5 A, 125 V fast-blow fuse.

Symptom:

- No marks or very light marks appear on the part.

Possible Causes:

- The part is not positioned at the point of focus.

Locate the final test report shipped with your marking head to determine the actual working distance of the lens. Measure the working distance from the bottom of the focusing lens mount to the surface of the part to be marked and ensure that it matches the distance specified on the final test report.

Symptom:

- Mark quality deteriorates gradually over time and/or laser power must be increased to maintain quality.

Possible Causes:

- The focusing lens is coated with marking debris or vapor.

Carefully inspect and clean the lens as described in the "Maintenance/Troubleshooting" section of your marking head's *Operator's Manual*.

Symptom:

- *ON* and *Status* indicators on the marking head are illuminated green, but no mark is created when the *Test Mark* pushbutton is pressed.

Possible Causes:

- The laser is not enabled.

Check that the laser's *Ready* light is illuminated. On Series 48 and Evolution lasers, check that the *Lase* indicator is glowing dimly (tickle mode) and that it appears brighter (lasing mode) during marking. On Firestar lasers, check that *RDY* (Ready) and *SHT* (Shutter) indicators are lit and that the *LASE* indicator illuminates during marking. Refer to your laser's *Operator's Manual* if indicators do not illuminate as described.

- The *Laser Control* cable is not connected between the laser and the marking head.

Refer to the "Getting Started" section of your marking head's *Operator's Manual* for information on connecting the *Laser Control* cable between the marking head and your particular laser.

- The Fast Acting Safety Interlock (FASI) feature on the Fiber Link Card is enabled.

If the FASI feature is enabled (DIP switch #5 on the Fiber Link Controller Card set to "On"), then a 15–40 V signal must be applied to *Input Bit 3* (*IN3*) on the FH/Fenix marking head before the laser will fire.

Note: On older ISA-bus Fiber Link Controller Cards, the FASI feature is controlled by DIP switch # 6. Refer to the *Operator's Manual* that came with your particular marking head for details.

Symptom:

- A Tracker marking head does not track moving parts.

Possible Causes:

- Tracking is not enabled.

Check that Track Marking Object on WinMark's *Tracking* tab is set to *Yes*.

- Encoder pulses are not being read by the FH/Fenix Tracker head.

Close WinMark Pro and open Digital Scope (DigScope.exe). Rotate the conveyor or position encoder and verify that *Input Bit 1* (*IN1*) and *Input Bit 2* (*IN2*) toggle as the encoder rotates.

- The part sensor signal is not being read by the FH/Fenix Tracker head.

Close WinMark Pro and open Digital Scope. Trigger the part sensor and verify that *Input Bit 0* (*IN0*) toggles.

- Part movement is not being sensed in the proper direction.

The encoder's "A" and "B" phases are inverted. Toggle Invert Encoder Direction on WinMark's *Tracking* tab.

Symptom:

- Characters or graphics within the mark are distorted or missing.

Possible Causes:

- Marking did not finish while the part was inside the marking field of the lens.

Ensure that the mark is positioned as close to the upstream edge of the *Drawing Canvas* as possible.

- Line speed is set too fast.

Lower line speed.

- The mark is too slow for the required line speed.

Lower mark cycle time by changing fonts, decreasing font size, or reducing line spacing. Simplify or eliminate graphic objects in the mark. Reduce graphic object size or resolution.

Symptom:

- Tracker marks are skewed or distorted.

Possible Causes:

- Encoder resolution is set improperly.

Load and mark the *linestack200.mkh* or *linestack370.mkh* file (located in the WinMark folder). Refer to the "Tracker" section in the marking head's *Operator's Manual* for instructions on how to fine-tune encoder resolution settings.

- Line speed is set too fast.

Lower line speed.

- The actual part motion vector is different than the value entered as the Motion Vector on WinMark's *Tracking* tab.

Check that actual direction of part motion under the marking head equals the entered Motion Vector value.

Automation

Symptom:

- The FH/Fenix marking head or WinMark Pro does not respond to automation inputs or the automation controller does not "see" marking head/WinMark Pro outputs.

Possible Causes:

- No Fiber Link Controller Card is selected.

In WinMark Pro, click the *Tools/General Settings/Applications Settings* tab. Scroll down to I/O Card Selection and verify that a card is selected. If no card is shown on the drop-down list, verify that a fiber link card is installed in your computer, and then check that drivers for the PCI-bus FLCC are properly installed (refer back to the FLCC troubleshooting section).

- I/O voltage levels are incorrect.

Review the "Technical Reference" section in your marking head's *Operator's Manual* to verify that I/O signals levels are in the correct voltage range of 15–40 VDC.

- I/O wiring is not properly connected to the marking head.

Double-check field wiring to ensure that input/output signals are routed to the correct pins and that their respective return (ground) paths are wired to the proper pins.

Use Digital Scope (Figure 21-2) to verify proper I/O functionality between the marking head/FLCC and the automation controller. Digital Scope's input "buttons" pop-in as each individual input is activated (when the correct voltage level is applied). Press an output "button" to activate an output.

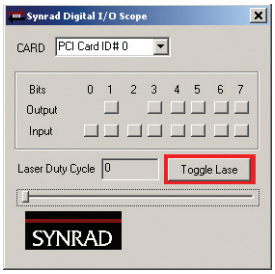


Figure 21-2 Digital Scope window

Note: To prevent I/O conflicts, always close WinMark Pro or WinMark Launcher before opening Digital Scope. Close Digital Scope before operating the marking head membrane panel pushbuttons or before starting WinMark Pro or WinMark Launcher.

- The Fiber Link card is incorrectly addressed (ISA-bus FLCCs only).

When using an ISA-bus FLCC, verify that the I/O address specified in WinMark Pro's *Event Builder* matches the current ISA card address. Remember to convert the hexadecimal card address to a decimal address for entry into WinMark Pro. The default hex address (330) equals 816 decimal.

Graphics

Vector graphics

Symptom:

- The vector import filter dialog box appears, but when the progress bar ends and the dialog closes there is no import object on the *Drawing Canvas* and no additional objects are listed in the *Object List*.

Possible Causes:

- The graphics file was created and saved using a newer version of software than is supported by WinMark's File Import Filters.

Open the graphics file using the graphics program that originally created it. Use the program's "Save As…" command to resave the file in an older format. Refer to the *SupportedFileFormats* document on the WinMark Pro web site or on the WinMark Pro CD for a list of compatible graphics formats and versions.

For example, you create an EPS graphics file using Adobe® Illustrator® version 12 (CS2). Current WinMark Pro File Import Filters only import EPS files created in versions 4.0 through 7.0. You must resave the file in an Illustrator version 7.0, or earlier, format in order to import the file.

Bitmap graphics

Symptom:

- Imported bitmap graphics are rescaled when imported into WinMark Pro.

Possible Causes:

- The graphics file was created and saved at a resolution other than the Resolution setting specified on the *Tools/General Settings…/Object Defaults* tab.

WinMark Pro does not resample bitmap images during importation. WinMark Pro "expects" imported bitmaps to match the Resolution setting shown on the *Tools/General Settings…/Object Defaults* tab; if not, then image size is increased or decreased accordingly—the change in image size is inversely proportional to the change in image resolution.

Figure 21-3 shows the results of importing three filled squares when WinMark's default Resolution is set to 600. Each square measured one inch by one inch when drawn and was separately imported into WinMark Pro in a .TIF (bitmap) format. The square on the left is saved at 600 DPI and imports as designed; it measures one-inch across. The center square is saved at 300 DPI. Since the resolution increased by a factor of two to 600 DPI during importation, image size decreased by a factor of two. This imported square now measures one-half inch on a side. The right-hand square illustrates the third case. This time, the one-inch square is saved at 900 DPI and then imported into WinMark Pro. Since image resolution decreased by 67% during importation (600/900), image size increased by a factor of 1.5 (1/.67). The imported square now measures one and one-half inches on a side.

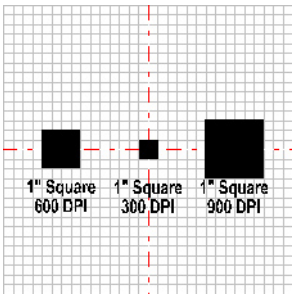


Figure 21-3 Bitmap image size/resolution comparison

Note: Older versions of WinMark Pro increased or decreased bitmap image size corresponding to a fixed bitmap resolution of 600 DPI.

Symptom:

- Color bitmap mark quality is not as good as expected.

Possible Causes:

- The Color Reduction Method property is not optimized.

Because laser marking (with the exception of the 8-bit *Grayscale* Color Reduction Method) is a monochrome—black and white or on/off—process, color or continuous-tone bitmaps are converted to 1-bit images when imported. This conversion is accomplished using *Bayer Dithering*, *Error Diffusion*, or *Halftone* algorithms. These algorithms create the illusion of color or continuous-tone by varying the size, shape, and/or spacing of the dots making up the image just as books, newspapers, and magazines do. WinMark Pro versions 4.0.0.4018 and above retain a copy of the original bitmap image in memory after importation. After changing bitmap Resolution and/or resizing the image, use the Color Reduction Method property (on the *Format* tab) to change the color reduction algorithm as required. Although WinMark Pro now displays a more accurate view of actual mark quality, test marking is still recommended.

Use the *Grayscale* Color Reduction Method to mark true 8-bit images. An 8-bit grayscale image provides 256 levels of gray ranging from 0 (white) to 255 (black). The *Grayscale* Color Reduction Method can reproduce fine images on "hard" surfaces or engrave "softer" materials.

Note: Prior to FH Series firmware version 17 and WinMark Pro build 4.3.3.4528, laser power control was limited to 7-bit resolution because Power settings were limited to integer values (1% steps). This has a drastic effect on grayscale marking because it changes the apparent resolution of the image. For example: 7-bit resolution equals 128 levels (0–127). If the mark requires a Power setting of 30% to obtain the desired mark quality, image resolution becomes 30% of 127, which equals 38.1. With integer Power resolution, the original 8-bit (256 level) grayscale image effectively consists of only 38 levels of gray.

In WinMark pro versions 4.3.3.4528 and above (with FH Series firmware versions 17 and above), FH heads support 8-bit Power resolution in 0.1% steps. In the example given above, a 30% Power setting times 255 (8-bit resolution) equals 76 levels of gray—twice the 7-bit value of 38 shades of gray.

- The Resolution property value is not correct.

When marking bitmap (raster) images, set a Resolution value that corresponds to the spot size of the focusing lens. For example, a 200 mm focusing lens with a spot size of 0.011" is capable of marking approximately 100 discrete microvectors per inch. As a general rule, set bitmap Resolution equal to 200 when using an 80 mm lens; 150 for a 125 mm lens; 100 for a 200 mm lens; and 50 when marking with a 370 mm lens. Higher Resolution values cause raster scan lines to overlap; the higher the Resolution, the greater the overlap. Start with the minimum specified Resolution for the selected lens and, if necessary, increase Resolution in multiples of the recommended values 2×, 3×, etc).

Symptom:

- When importing a metafile (a graphics format containing both vector and raster graphics) into WinMark Pro the last raster object to import is "garbled".

Possible Causes:

- An incompatibility exists with Windows® 2000 and Windows® XP operating systems.

Edit the graphics file so that vector and bitmap objects import separately.

ActiveMark

Symptom:

- The ActiveX® application appears to "freeze" after the MarkDrawing method is called.

Possible Causes:

- Input automation commands are specified in the .mkh mark file.

Performing input automation within the mark file may prevent the ActiveX application program from regaining control of the operating system if the MarkDrawing method is called and input signals are not available. Set up input automation functions within the ActiveX application (not in the .mkh file) so that your application or user-interface remains in charge of all pre-mark and post-mark automation. WinMark Pro's MarkDrawing method should only maintain control of the operating system long enough to control laser on/off commands during the actual marking process.

Symptom:

- The WinMark Pro ActiveX control is not available after upgrading to a newer version of WinMark (Microsoft® Visual C++® only).

Possible Causes:

- WinMark Pro's ActiveX control was not updated in the MS Visual C++ project.

After installing or upgrading WinMark Pro, you must add (or update) WinMark's ActiveX control in the Visual C++ project. See *Adding ActiveX controls.pdf* on the WinMark Pro CD or on the WinMark web site for instructions on adding or updating ActiveX controls in Visual C++. Remember that the C++ application must be built to include the new control changes.

Laser Marking FAQ

SYNRAD's *Laser Marking FAQ* is available for downloading or browsing from the WinMark Pro web site at http://www.winmark.com. You can also access the *Laser Marking FAQ* directly from WinMark by clicking *Help\Synrad WinMark FAQ*. Our Frequently Asked Questions database answers many common marking questions in categories including *General troubleshooting*, *WinMark Pro automation*, *Tracking operation*, *ActiveX*, and others. The *Laser Marking FAQ* also includes links to various SYNRAD technical bulletins.

Resources

See the *Release Notes* provided with each build of WinMark Pro for the latest information on new features, bug fixes, and miscellaneous issues.

Your SYNRAD Regional Sales Manager can provide reference materials including Outline & Mounting drawings, Operator's Manuals, Technical Bulletins, Application Newsletters, and much more. Most of these materials are also available directly from SYNRAD's web site at http://www.synrad.com, the WinMark Pro web site at http://www.winmark.com, or from the WinMark Pro CD.